

**SIACC (SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN DE  
CONJUNTOS CERRADOS)**

**Yina Paola Chaves Silva.  
Miguel Angel Colorado Perez.**

**UNIVERSIDAD LIBRE.  
FACULTAD DE INGENIERIA  
PROGRAMA DE INGENIERIA DE SISTEMAS  
BOGOTA DC.  
OCTUBRE 2012.**

**SIACC (SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN DE  
CONJUNTOS CERRADOS)**

**AUTORES:**

**YINA PAOLA CHAVES SILVA.**

**MIGUEL ANGEL COLORADO PEREZ.**

**PROYECTO DE GRADO PARA OPTAR ALA TITULO DE INGENIERO DE  
SISTEMAS E INFORMATICA.**

**UNIVERSIDAD LIBRE.**

**FACULTAD DE INGENIERIA**

**PROGRMA DE INGENIERIA DE SISTEMAS**

**BOGOTA DC.**

**OCTUBRE 2012.**

## Nota de Aceptación

---

---

---

---

---

---

---

**Jurado**

---

**Jurado**

**Bogotá D.C., Agosto 28 del 2012**

## **DEDICATORIA**

Este proyecto se lo dedico principalmente a Dios por tenerme con vida y darme la sabiduría para lograr esta meta, a mi mama Teresa Pérez por darme la vida y enseñarme a ser el hombre que hoy en día soy, con dedicación y cariño; a mi papá José Oseas Colorado por su gran apoyo y confianza para estudiar Ingeniería de sistemas. A mi familia por su compañía y creer en mi durante este tiempo y a Paola Chaves por estar a mi lado y ser mi inspiración en este proyecto y decisión.

Miguel Ángel Colorado Pérez

La dedicatoria de mi proyecto de grado quiero que en primer lugar sea a Dios porque me dio la vida, la sabiduría, las capacidades y la fortaleza para lograr esta meta; en segundo lugar lo dedico a mi familia por el apoyo que me dio con mi estudio, comprensión, vivencias y en cada instante con sus enseñanzas y compañía, les doy gracias a Teresa Piñeros, Luz Mery Silva y Jairo Chaves por el apoyo que me brindan cada instante de la vida porque así he logrado lo que hoy en día soy y doy a los demás. Gracias a Miguel Ángel Colorado el cual es mi principal apoyo y colaborador no solo de este gran proyecto sino también en el proyecto de vida.

Yina Paola Chaves Silva

## **AGRADECIMIENTOS**

Gracias a DIOS por tenernos en este mundo con un fin y logrando nuestras metas y sueños junto a nuestras familias y a los maestros de la universidad que hicieron en nuestras vidas un deseo posible.

Agradecemos la colaboración para lograr esta idea al ingeniero Giovanny Briceño por sus conocimientos, disciplina, colaboración, tiempo y dedicación a nosotros como estudiantes y como personas a cargo de un proyecto, viendo el fruto desde sus clases hasta sus tiempos libres para culminar satisfactoriamente este proyecto. Al ingeniero Álvaro Rojas Daza por inculcarnos desde varias perspectivas la importancia de ser ingenieros de sistemas. Al ingeniero Gustavo Rivera por sus conocimientos e interés en el lenguaje JAVA y sus aplicaciones.

Agradecemos a nuestras familias por su ayuda económica con gran esfuerzo para poder hoy lograr este gran pasó en nuestras vidas.

## TABLA DE CONTENIDO

### Contenido

LISTA DE IMAGENES .....	9
LISTA DE TABLAS .....	10
GLOSARIO .....	11
RESUMEN .....	16
INTRODUCCIÓN .....	17
1. PLANTEAMIENTO DEL PROBLEMA .....	18
1.1 FORMULACIÓN DEL PROBLEMA (PREGUNTA) .....	18
1.2. JUSTIFICACIÓN.....	18
1.3. OBJETIVOS: .....	19
1.3.1. GENERAL .....	19
1.3.2. ESPECÍFICOS .....	19
1.4. ALCANCE.....	20
1.5. DESCRIPCIÓN DEL SISTEMA: .....	21
1.6. HIPÓTESIS .....	22
2. DISEÑO METODOLÓGICO.....	23
2.1. TIPO DE INVESTIGACIÓN. ....	23
2.2. UNIVERSO, MUESTRA, DISEÑO DE VARIABLES.....	23
2.3. FUENTES DE INFORMACIÓN.....	24
2.4. INSTRUMENTOS Y ANÁLISIS.....	24
2.5. ANALISIS DE INFORMACION .....	24
2.5.1. Resultados de las encuestas a propietarios. ....	24
2.6. METODOLOGÍA.....	28
3. MARCO REFERENCIAL.....	28
3.1. MARCO HISTÓRICO .....	28
3.1.1. Diagnostico del estado actual:.....	30
3.1.2. Otros sistemas similares: .....	31
3.2. MARCO TEÓRICO .....	31
3.2.1. Sistemas de información. ....	31

3.2.2. Ingeniería de software: .....	32
Fuente: Autores.....	33
3.2.3. Conceptos de desarrollo de software: .....	33
3.2.4. Bases de datos:.....	34
• Control sobre la redundancia de datos: .....	35
• Consistencia de datos:.....	35
• Compartición de datos: .....	35
• Mantenimiento de estándares:.....	36
• Mejora en la integridad de datos:.....	36
• Mejora en la seguridad: .....	36
• Mejora en la accesibilidad a los datos: .....	36
• Mejora en la productividad:.....	36
• Mejora en el mantenimiento:.....	37
• Aumento de la concurrencia: .....	37
• Mejora en los servicios de copias de seguridad: .....	37
• Complejidad:.....	37
• Coste del equipamiento adicional: .....	38
• Vulnerable a los fallos:.....	38
• Tipos de Campos.....	38
3.2.5. Servidores WEB: .....	41
3.2.6. Metodología.....	41
RUP.....	41
3.2.7. SCRUM .....	43
3.3. MARCO CONCEPTUAL.....	45
3.3.1. Frameworks.....	45
3.3.2. JUDE (Java and UML Developers Environment) .....	45
3.3.3. IDE Netbeans .....	46
3.3.4. Stakeholder .....	47
3.3.5. Tomcat .....	47
3.3.6. UML.....	48
3.3.7. Workflow.....	48
3.3.8. XML (eXtensibleMarkupLanguage) .....	48

3.4 MARCO LEGAL.....	49
3.4.1. La ley 675 de 2001, de la propiedad horizontal cita: .....	49
3.5. MARCO TECNOLÓGICO.....	52
3.5.1. Recursos Hardware para el desarrollo del proyecto:.....	52
3.5.2. Recursos Software para el desarrollo del proyecto: .....	52
3.5.3. Ambiente de instalación: .....	53
4. ESTRUCTURA TEMÁTICA – DESARROLLO DEL PROYECTO .....	54
4.1. CAPITULO 1: ANÁLISIS.....	54
4.1.1. Descripción general del sistema.....	54
4.1.2. Requerimientos Funcionales. ....	54
4.1.3. REQUERIMIENTOS NO FUNCIONALES. ....	73
4.1.4. CRONOGRAMA.....	73
4.2. CAPITULO 2 DISEÑO: .....	75
4.2.1. Diseño de arquitectura .....	75
4.2.2. Diseño de paquetes.....	76
4.2.3. Diseño de clases .....	76
4.2.4. Diseño de pruebas .....	79
4.3. CAPITULO 3. IMPLEMENTACIÓN.....	82
4.3.1. DIAGRAMA DE MER. ....	83
4.3.2. Estándares y prototipos de las pantallas del sistema. ....	84
4.3.3. RESULTADO DE PRUEBAS .....	95
4.4. CAPITULO 4. IMPLANTACIÓN.....	96
4.4.1. INSTALACIÓN. ....	96
5. RESULTADOS Y DISCUSIÓN. ....	97
6. Conclusiones. ....	98
7. Recomendaciones. ....	99
8. Bibliografía. ....	100
9. Anexos .....	104



## LISTA DE IMAGENES

IMAGEN 1. DESCRIPCIÓN DEL SISTEMA.....	21
IMAGEN 2. PREGUNTA1.....	25
IMAGEN 3. PREGUNTA2.....	25
IMAGEN 4. PREGUNTA3.....	26
IMAGEN 5. PREGUNTA4.....	26
IMAGEN 6. FASES DE LA INGENIERÍA DE SOFTWARE.....	30
IMAGEN 7. EJEMPLO DE MER.....	37
IMAGEN 8. DIAGRAMA DE UN SERVIDOR WEB.....	38
IMAGEN 9. CICLO DE VIDA DE RUP.....	40
IMAGEN 10. IMAGEN SCRUM.....	42
IMAGEN 11. ARQUITECTURA DEL SISTEMA.....	71
IMAGEN 12. DISEÑO DE PAQUETES.....	72
IMAGEN 13. DIAGRAMA DE CLASES GENERAL.....	72
IMAGEN 14. DIAGRAMA DE CLASES ENTITY.....	73
IMAGEN 15. DIAGRAMA DE CLASES BEANS.....	74
IMAGEN 16. DIAGRAMA DE CLASES CONTROLER.....	75
IMAGEN 17. MODELO ENTIDAD RELACIÓN.....	76
IMAGEN 18. PANTALLA DE CARGUE.....	77
IMAGEN 19. PANTALLA DEL LOGIN.....	77
IMAGEN 20. PANTALLA DEL PERFIL DEL ADMINISTRADOR.....	78
IMAGEN 21. PANTALLA DE LAS NOTICIAS EN ADMINISTRADOR.....	78
IMAGEN 22. PANTALLA DE EVENTOS EN ADMINISTRADOR.....	79
IMAGEN 23. PANTALLA DE CORREO EN ADMINISTRADOR.....	79
IMAGEN 24. PANTALLA DE SALÓN COMUNAL EN ADMINISTRADOR.....	80
IMAGEN 25. PANTALLA INFORMES EN ADMINISTRADOR.....	80
IMAGEN 26. PANTALLA CARTELERA SOCIAL EN ADMINISTRADOR.....	81
IMAGEN 27. PANTALLA MODIFICAR PROPIETARIO EN ADMINISTRADOR.....	81
IMAGEN 28. PANTALLA CERRAR SESIÓN EN ADMINISTRADOR.....	82
IMAGEN 29. PANTALLA DATOS DEL PERFIL EN PROPIETARIO.....	82
IMAGEN 30. PANTALLA DE NOTICIAS EN PROPIETARIO.....	82
IMAGEN 31. PANTALLA DE EVENTOS EN PROPIETARIO.....	83
IMAGEN 32. PANTALLA DE CORREO EN PROPIETARIO.....	83
IMAGEN 33. PANTALLA DE SALÓN COMUNAL EN PROPIETARIO.....	84
IMAGEN 34. PANTALLA DE INFORMES EN PROPIETARIO.....	84
IMAGEN 35. PANTALLA DE CARTELERA SOCIAL EN PROPIETARIO.....	85
IMAGEN 36. PANTALLA DE PERFIL EN PROPIETARIO.....	85

## LISTA DE TABLAS

TABLA1: MARCO TECNOLÓGICO.....	49
TABLA2: CRONOGRAMA.....	70

## GLOSARIO

**AJAX:** Acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.<sup>1</sup>

**API:** Interfaz de programación de aplicaciones (IPA) o API (del inglés *Application Programming Interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas vulgarmente "librerías").<sup>2</sup>

**BEANS:** *Abreviatura científica del botánico William Jackson Bean 1863-1947*

Un Bean es un componente software que tiene la particularidad de ser reutilizable y así evitar la tediosa tarea de programar los distintos componentes uno a uno. Se puede decir que existen con la finalidad de ahorrarnos tiempo al programar. Es el caso de la mayoría de componentes que manejan los editores visuales más comunes. Los que hayan utilizado Visual Studio, Eclipse o Delphi por ejemplo, ya estarán familiarizados con ellos.<sup>3</sup>

**DAO:** En software de computadores, un Data Access Object (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object.<sup>4</sup>

**FACELETS:** Facelets es un lenguaje de declaración de páginas, poderoso pero ligero, que es usado para construir vistas de JavaServer Faces usando plantillas de estilo de HTML y construyendo arboles de componentes. Las características que Facelets incluyen son algunas de las siguientes:

- Uso de XHTML para crear páginas web
- Soporte para librerías de etiquetas Facelets que se suman a las librerías de JavaServer Faces y JSTL
- Soporte para el Lenguaje de Expresiones (EL por su siglas en ingles)

---

<sup>1</sup>**AJAX** [En línea] disponible en internet URL: <http://es.wikipedia.org/wiki/AJAX>

<sup>2</sup>**API** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Interfaz\\_de\\_programaci%C3%B3n\\_de\\_aplicaciones](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones)

<sup>3</sup>**BEANS** [En línea] disponible en internet URL: <http://es.wikipedia.org/wiki/Bean>

<sup>4</sup>**DAO** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Data\\_Access\\_Object](http://es.wikipedia.org/wiki/Data_Access_Object)

- Plantillas para componentes y páginas.<sup>5</sup>

**FRAMEWORKS:** La palabra inglesa "framework" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de *software*, un *framework* o *infraestructura digital*, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, con base a la cual otro proyecto de *software* puede ser más fácilmente organizado y desarrollado.<sup>6</sup>

**GUI:** La interfaz gráfica de usuario, conocida también como GUI (del inglés *graphical user interface*) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.<sup>7</sup>

**HTML:** Siglas de *HyperTextMarkupLanguage* («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.<sup>8</sup>

**JAVA:** Es un lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. El lenguaje en sí mismo toma mucha de su sintaxis de C, Cobol y Visual Basic, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo

---

<sup>5</sup>**FACELETS** [En línea] disponible en internet URL:

<http://programacionconejemplos.blogspot.com/2011/08/que-es-facelets-52.html>

<sup>6</sup>**FRAMEWORKS** [En línea] disponible en internet URL: <http://es.wikipedia.org/wiki/Framework>

<sup>7</sup>**GUI** [En línea] disponible en internet URL:

[http://es.wikipedia.org/wiki/Interfaz\\_gr%C3%A1fica\\_de\\_usuario](http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario)

<sup>8</sup>**HTML** [En línea] disponible en internet URL: <http://www.ri5.com.ar/ayuda03.php>

para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.<sup>9</sup>

**JPA:** Java Persistence API, más conocida por sus siglas JPA, es la API de persistencia desarrollada para la plataforma Java EE

Es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones usando la Plataforma Java en sus ediciones Standard (Java SE) y Enterprise (Java EE).

Persistencia en este contexto cubre tres áreas:

- La API en sí misma, definida en `javax.persistence.package`
- La Java Persistence Query Language (JPQL)
- Metadatos objeto/relacional

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional), como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs).<sup>10</sup>

**JSF:** JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServerPages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL.

JSF incluye:

- Un conjunto de Apis para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para JavaServerPages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados.<sup>11</sup>

**JSP:** JavaServerPages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSPs permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas

---

<sup>9</sup> **JAVA** [En línea] disponible en internet URL:

[http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

<sup>10</sup> **JPA** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Java\\_Persistence\\_API](http://es.wikipedia.org/wiki/Java_Persistence_API)

<sup>11</sup> **JSF** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/JavaServer\\_Faces](http://es.wikipedia.org/wiki/JavaServer_Faces)

etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o TagLibraries) externas e incluso personalizadas.<sup>12</sup>

**MVC:** Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.<sup>13</sup>

**PERSISTENCIA:** Se entiende por persistencia (en programación) como la acción de preservar la información de un objeto de forma permanente (guardar), pero a su vez también se refiere a poder recuperar la información del mismo (leer) para que pueda ser nuevamente utilizada.<sup>14</sup>

**POSTGRESQL:** Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global DevelopmentGroup*).<sup>15</sup>

**PROPIEDAD HORIZONTAL:** La propiedad horizontal es una institución jurídica que hace alusión al conjunto de normas que regulan la división y organización de diversos inmuebles, como resultado de la segregación de un edificio o de un terreno común.

En sí la propiedad horizontal no es un bien inmueble en particular sino un régimen que reglamenta la forma en que se divide un bien inmueble y la relación entre los propietarios de los bienes privados y los bienes comunes que han sido segregados de un terreno o edificio. La propiedad horizontal permite la organización de los copropietarios y el mantenimiento de los bienes comunes.<sup>16</sup>

**RICHFACES:** Es una biblioteca de código abierto basada en Java que permite crear aplicaciones web con Ajax.

Construye sobre el framework de Java Server Faces. Sobre él, implementa unos filtros para permitir peticiones Ajax en la página. La singularidad del planteamiento

---

<sup>12</sup> **JSP** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/JavaServer\\_Pages](http://es.wikipedia.org/wiki/JavaServer_Pages)

<sup>13</sup> **MVC** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

<sup>14</sup> **PERSISTENCIA** [En línea] disponible en internet URL:

[http://es.wikipedia.org/wiki/Persistencia\\_de\\_objetos](http://es.wikipedia.org/wiki/Persistencia_de_objetos)

<sup>15</sup> **POSTGRESQL** [En línea] disponible en internet URL: <http://es.wikipedia.org/wiki/PostgreSQL>

<sup>16</sup> **PROPIEDAD HORIZONTAL** [En línea] disponible en internet URL:

[http://es.wikipedia.org/wiki/Propiedad\\_horizontal](http://es.wikipedia.org/wiki/Propiedad_horizontal)

que ofrece es que la petición Ajax provoca una ejecución en el servidor y finalmente una renderización parcial o total de la página del navegador. Así, el control de lo que sucede está en el servidor.<sup>17</sup>

**SERVLET:** Son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej.: Tomcat) y extienden su funcionalidad.

La palabra *servlet* deriva de otra anterior, *applet*, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web. Por contraposición.

El uso más común de los *servlets* es generar todas páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.<sup>18</sup>

**SOLUCIÓN:** El término solución, del latín *solutio*, tiene dos grandes usos. Por un lado, se trata de la acción y efecto de resolver una dificultad o una duda.<sup>19</sup>

**WEB SERVER:** Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.<sup>20</sup>

**XML:** Siglas en inglés de *eXtensibleMarkupLanguage* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información. (Bases de datos Silberschatz).<sup>21</sup>

---

<sup>17</sup>**RICHFACES** [En línea] disponible en internet URL: <http://es.wikipedia.org/wiki/RichFaces>

<sup>18</sup>**SERVLET** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Java\\_Servlet](http://es.wikipedia.org/wiki/Java_Servlet)

<sup>19</sup>**SOLUCION** [En línea] disponible en internet URL: <http://definicion.de/solucion/>

<sup>20</sup>**WEB SERVER** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web)

<sup>21</sup>**XML** [En línea] disponible en internet URL: [http://es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://es.wikipedia.org/wiki/Extensible_Markup_Language)

## **RESUMEN**

SIACC (Sistema de Información para la Administración de Conjuntos Cerrados), es un sistema el cual facilita la comunicación entre residentes y administrador dando solución a la problemática de la falta de interacción entre los actores, permite esta acción recíproca estando dentro o fuera del conjunto y sin tener en cuenta el horario, porque la herramienta deja plasmada la información para descargarla y consultarla cada vez que entren los usuarios, sin necesidad de acercarse a la administración, permite ver las noticias, eventos, cartelera social, fechas reservadas y disponibles del salón comunal, informes contables, información personal y de las empresas que prestan los servicios de aseo y vigilancia en el conjunto.



## INTRODUCCIÓN

Este proyecto se crea pensando en los residentes los cuales no cuentan con mucho tiempo para poder involucrarse totalmente en las decisiones o consultas con respecto a los procesos que se lleva en el conjunto residencial donde viven, ya que por el trabajo los residentes salen en altas horas de la mañana y llegan a altas horas de la noche, donde se evidencia que es el mismo horario del administrador. La solución es la de facilitar la comunicación e interacción entre el residente y el administrador, teniendo en cuenta todos los procesos en que ambos intervienen, como los son los informes, cartelera social, eventos, noticias y demás las cuales son importantes en la convivencia de un conjunto residencial.

Por consiguiente se crea una herramienta Web desarrollada en JAVA con librería de componentes visuales Richfaces y arquitectura MVC, la cual es amigable y sencilla de trabajar tanto para el administrador como los residentes, se llega a los involucrados dando a conocer la parte de interacción social del conjunto residencial, donde cuenta con esta flexible comunicación desde cualquier lugar de trabajo sin estar limitados a los horarios disponibles de los actores.

## **1. PLANTEAMIENTO DEL PROBLEMA.**

Actualmente en la administración de los conjuntos cerrados de la localidad 10 de Engativá, barrios Cortijo y Ciudadela Colsubsidio no tienen toda la información en medios digitales, los registros son realizados manualmente y no existe un control sobre las empresas que prestan servicios en dicho conjunto.

En general la administración es principalmente gestionada por una persona, que tiene compromiso y responsabilidad social, económica, ética con el conjunto, las cuales son expuestas en el anexo1, sin una ayuda para lograr eficiencia al realizar sus labores.

La administración dentro de sus funciones debe convocar a diferentes eventos y actividades, pero, la comunidad no participa activamente por diversos inconvenientes, como lo son la disponibilidad de tiempo. El acercamiento de ambas partes es complicada al no existir otros medios de comunicación a diferencia del físico.

### **1.1 FORMULACIÓN DEL PROBLEMA (PREGUNTA)**

¿El contar con una herramienta de software en un conjunto, permitirá apoyar y mejorar la gestión administrativa?

### **1.2. JUSTIFICACIÓN**

En la administración de los conjuntos cerrados se requiere centralizar la Información, para el administrador, los propietarios, el consejo de administración, el comité de convivencia, el contador y el revisor fiscal. Mejorando así la eficiencia del servicio de administración a los propietarios.

Teniendo en cuenta las falencias de seguridad que se dan en los conjuntos por el cambio de personal de las empresas prestadoras de servicios, se busca mejorar el control sobre los empleados por medio del aplicativo.

Por otro lado la comunicación entre la administración y los propietarios debe ser clara y permanente, para no desconocer las decisiones autónomas de la asamblea y la administración; por medio de la publicación de las noticias, los eventos y un

chat.

De acuerdo con la información suministrada anteriormente en la descripción del problema, es necesario suplir los requerimientos por medio de los módulos diseñados en la solución software. También se necesita automatizar y organizar la información de los conjuntos cerrados de los barrios Cortijo y Ciudadela Colsubsidio para brindar efectividad a los propietarios (los cuales son los directos clientes) y una mejor calidad en el servicio de administración, también logrando apoyar la gestión del administrador.

Este proyecto se justifica sobre la reducción de tiempos y costos en el que el administrador realiza el trabajo de la organización de la información y actividades, como el pago de las cuotas que el propietario hace, solicitud de préstamos de salón comunal, quejas y reclamos, publicación de anuncios en cartelera social, que el sistema podrá organizar en respuesta de la administración. Lo anterior se debe a que el horario del administrador de un conjunto es de medio tiempo y flexible, se debe tener en cuenta los imprevistos que pueden surgir en cualquier momento por parte de los propietarios.

Al finalizar el aplicativo web y móvil de administración de conjuntos cerrados, se mejorara la eficiencia del trabajo de la administración reduciendo costos de impresión de informes, documentos, reuniones extraordinarias, recibos y publicidad en la cartelera de información.

### **1.3. OBJETIVOS:**

#### **1.3.1. GENERAL**

General: Desarrollar un aplicativo web para facilitar la gestión administrativa por medio de un servicio web para mejorar la comunicación a través de las diversas funcionalidades del sistema y así lograr un optimo servicio entre el administrador y los residentes de los conjuntos cerrados de la localidad de Engativá en los barrios Cortijo y Ciudadela.

#### **1.3.2. ESPECÍFICOS**

- ✓ Definir y caracterizar los servicios administrativos comunes de conjuntos cerrados de la localidad 10 de los barrios Cortijo y Ciudadela Colsubsidio a través de encuestas a residentes y entrevistas a administradores de dichos conjuntos.
- ✓ Diseñar los módulos del aplicativo web, que facilitan la gestión en los conjuntos cerrados por medio de la caracterización de los servicios administrativos, como lo

son los eventos sociales, las noticias, el alquiler de áreas comunes, el control de residentes del conjunto, el registro de las empresas prestadoras de servicios y la visualización de documentos de administración.

Realizar comunicación de la administración con los residentes a través de un aplicativo web, con informes de control administrativo, reuniones extraordinarias de asamblea e información de actividades.

✓ Desarrollar un aplicativo web para la administración de conjuntos cerrados tomando la idea de la problemática de los conjuntos cerrados de la localidad 10 de los barrios Cortijo y Ciudadela.

#### **1.4. ALCANCE.**

El alcance de este proyecto es el realizar la catalogación de las funcionalidades de una herramienta Web, con servicios para acceder desde cualquier lugar o a través de dispositivos móviles con tecnologías http; que apoye la administración de conjuntos cerrados. La recolección de la información será a través de encuestas y entrevistas a los principales actores del sistema.

Se entregará el aplicativo web y los manuales de usuario, manual del producto y la documentación establecida para cada fase del proyecto.

Al hacer la entrega y pruebas, se dará a conocer la mejora del servicio de administración. Podemos ofrecerlo para implementarlo en otros conjuntos cerrados del sector.

El sistema ayudara a la gestión de administración, en los siguientes módulos y funcionalidades propuestos:

\* Social:

- Correo
- Noticias
- Eventos
- Cartelera social

\* Control del los apartamentos y residentes:

- Registro de residentes
- Actualización de residentes
- Actualización de administrador

\* Préstamo de espacios comunes:

- Reserva del salón comunal
- Tabla de reservas
- Eliminación de reservas

\* Informes Contables:

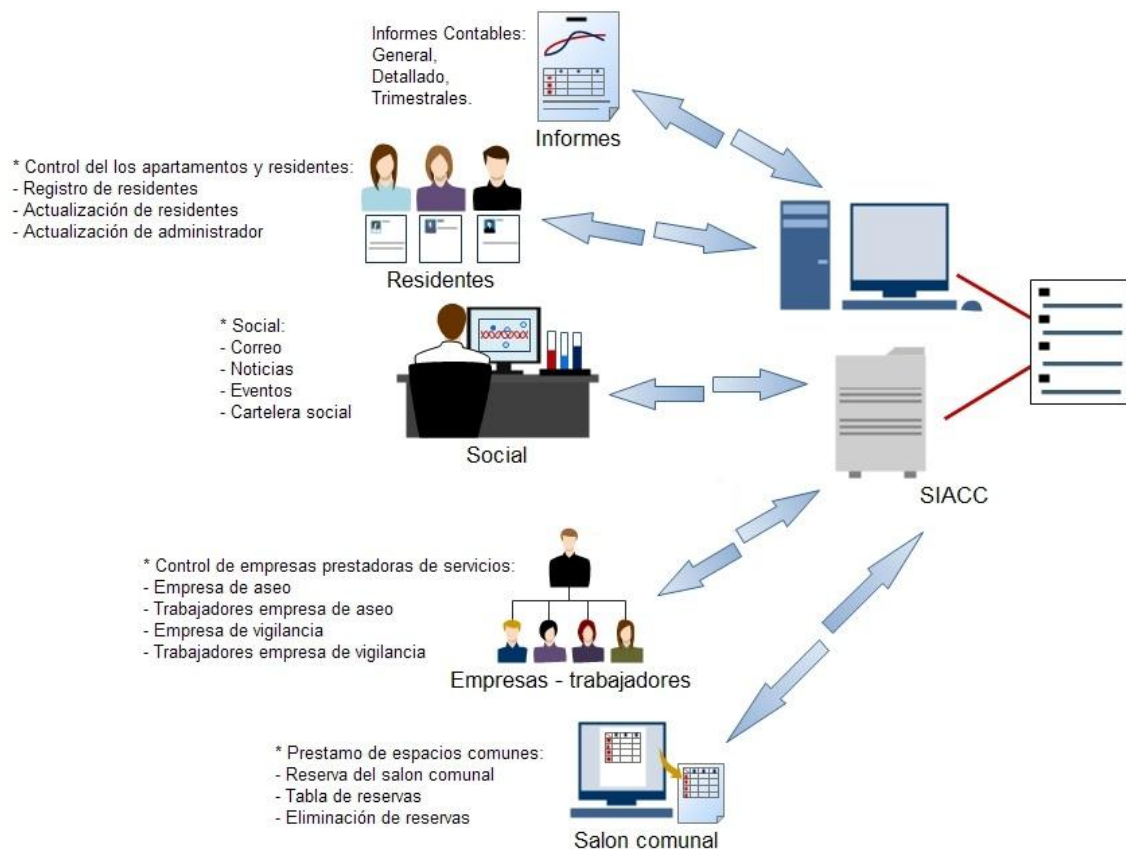
- General
- Detallado
- Trimestrales

\* Control de empresas prestadoras de servicios:

- Empresa de aseo
- Trabajadores empresa de aseo
- Empresa de vigilancia
- Trabajadores empresa de vigilancia

## 1.5. DESCRIPCIÓN DEL SISTEMA:

IMAGEN 1. DESCRIPCIÓN DEL SISTEMA



Fuente: Autores

**Imagen 1. Descripción del sistema:** En esta imagen se visualiza la propuesta Web que fundamenta SIACC. Se ve la generación de los informes, se tiene control

de los residentes del conjunto, en la parte social hay un correo que permite una comunicación más pronta entre el residente y el administrador, está el espacio para revisar las noticias, eventos y cartelera social la cual es la información que el administrador muestra a los residentes para tenerlos informados, se tiene el control de la información de las empresas prestadoras de servicios en el conjunto, y el prestamos del salón comunal.

Este prototipo del sistema se probara al final del proyecto en uno de los Conjuntos, en el barrio Cortijo o Ciudadela Colsubsidio, como conjunto piloto ó de referencia para los otros 4 de la zona dicha.

## **1.6. HIPÓTESIS**

El contar con una herramienta software en un conjunto cerrado permitirá mejorar y apoyar la gestión administrativa, porque se mejorara la comunicación entre las dos partes, los propietarios y la administración. Esto lo basamos en las encuestas y entrevista (Anexo 1. y Anexo 2.) realizadas a los propietarios y administrador de conjuntos cerrados respectivamente, anexos de resultados al final del documento. Las encuestas a los propietarios arrojaron: (Anexo3.)

El 62% de los propietarios usan diariamente internet, teniendo en cuenta que en general un 82% usa internet.

El 67% de los propietarios se comunican con la administradora personalmente y el 18% lo hace por medios no físicos como el teléfono (10%) y correo electrónico (8%).

La comunidad manifestó que si actualmente existiera un sistema de gestión de conjuntos cerrados lo usarían con una puntuación del 76%.

La acogida del sistema en un futuro se vio evidenciada en el 74% de los propietarios, porque pueden usarlo desde cualquier lugar, su sitio de trabajo, si con arrendadores desde su vivienda o cualquier otro lugar donde tenga acceso a internet.

El 8% de los propietarios que utilizan medios electrónicos para la comunicación con la administración los queremos incrementar teniendo en cuenta el porcentaje de los propietarios que utilizan internet diariamente y los que en un futuro utilizarían el sistema.

## **2. DISEÑO METODOLÓGICO.**

### **2.1. TIPO DE INVESTIGACIÓN.**

Nuestro Tipo de investigación es cuantitativa aplicada, basándonos en el concepto de la teoría general de los sistemas.

Es un tipo de investigación cuantitativa aplicada, porque analizaremos los procesos del sistema actual del conjunto cerrado, explicando cada uno y su función. Para este proyecto se usará la metodología RUP para los puntos en los cuales se define el análisis y diseño y SCRUM en el desarrollo e implementación. Este proceso se llevará a cabo por medio de encuestas (Anexo 1.) a los propietarios de 5 conjuntos y entrevistas (Anexo 2.) a 5 administradores, para determinar los módulos del aplicativo Web.

La teoría general de los sistemas nos permite entender al conjunto de elementos interrelacionados como un todo, teniendo como enfoque de observación al sistema social del conjunto cerrado, el cual es compuesto por personas que organizadas actúan como elementos que sirven a sus intereses mutuos y objetivos comunes. De esta manera podemos estudiar las interrelaciones que existen y las interdependencias para determinar los operadores funcionales que mejorarían la comunicación entre los diferentes elementos de nuestro sistema objetivo.

### **2.2. UNIVERSO, MUESTRA, DISEÑO DE VARIABLES.**

- **Universo:** Sistemas sociales de viviendas con espacios y actividades comunes que tienen como fin el garantizar la seguridad y convivencia pacífica en los inmuebles sometidos y la función social de la propiedad.

Como lo menciona la ley de propiedad horizontal un conjunto es “El desarrollo inmobiliario conformado por varios edificios levantados sobre uno o varios lotes de terreno, que comparten, áreas y servicios de uso y utilidad general, como vías internas, estacionamientos, zonas verdes, muros de cerramiento, porterías, entre otros.”

- **Muestra:** Tomaremos como muestra estratificada algunos de los conjuntos cerrados de la localidad 10 de los barrios Cortijo y Ciudadela, para utilizarlo como conjunto piloto de la aplicación.

## 2.3. FUENTES DE INFORMACIÓN.

- **Primarias:** La información se basa en las posibles recomendaciones y funcionalidades aportadas por administradores propietarios y/o residentes de conjuntos cerrados de la muestra seleccionada anteriormente.
- **Secundarias:** Ley de propiedad horizontal LEY 675 DE 2001
- **Terciarias:** Son las funciones del administrador, los derechos y obligaciones de los propietarios teniendo en cuenta el manual de convivencia y la ley de propiedad horizontal.

## 2.4. INSTRUMENTOS Y ANÁLISIS.

Las modalidades para el tratamiento de la información necesaria para la visualización son las encuestas a propietarios y/o residentes y las entrevistas a los administradores de los conjuntos cerrados. Y también para su visualización y mejor comprensión de esta información se manejará de forma estadística para la toma de las muestras evidenciada en el **Anexo 1 Encuesta a propietarios.**

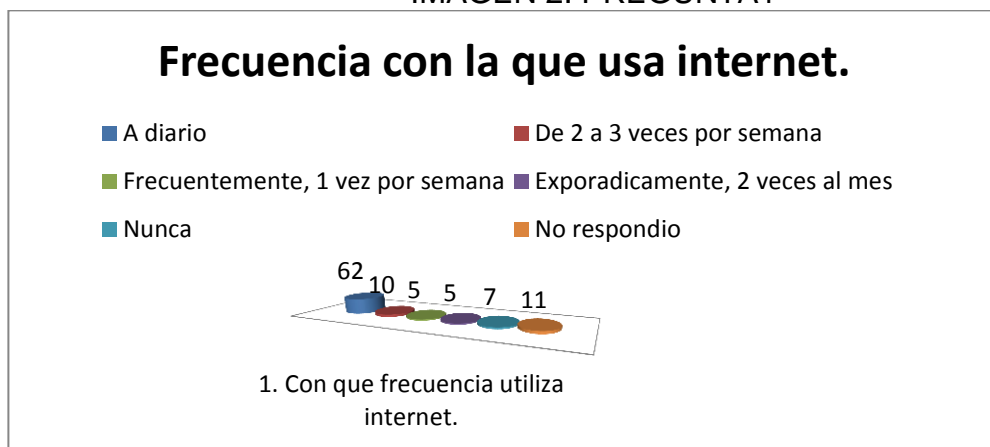
## 2.5. ANALISIS DE INFORMACION

### 2.5.1. Resultados de las encuestas a propietarios.

1. Con que frecuencia utiliza internet.	
A diario	62
De 2 a 3 veces por semana	10
Frecuentemente, 1 vez por semana	5
Esporádicamente, 2 veces al mes	5
Nunca	7
No respondió	11
Total:	100



IMAGEN 2. PREGUNTA1

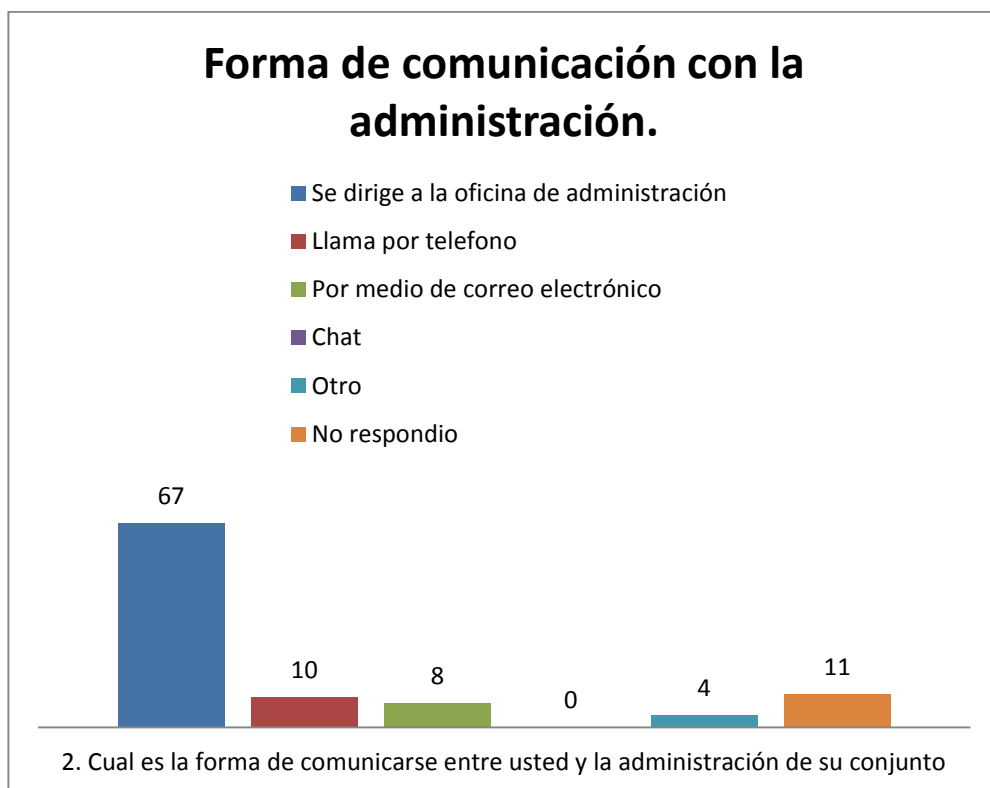


Fuente: Autores

**Imagen 2. Pregunta uno:** En la imagen se visualiza el grafico generado de la pregunta uno de la encuesta realizada a los residentes.

2.Cuál es la forma de comunicarse entre usted y la administración de su conjunto	
Se dirige a la oficina de administración	67
Llama por teléfono	10
Por medio de correo electrónico	8
Chat	0
Otro	4
No respondió	11
Total	100

IMAGEN 3. PREGUNTA2



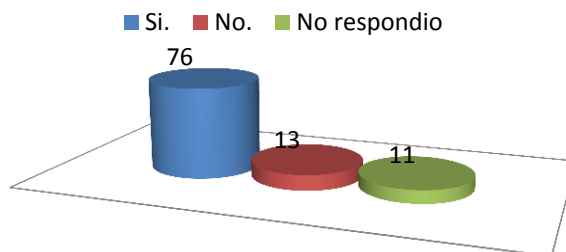
Fuente: Autores

**Imagen 3. Pregunta dos:** En la imagen se visualiza el grafico generado de la pregunta dos de la encuesta realizada a los residentes.

3. Si el conjunto tuviera una forma de comunicación WEB, la usaría	
Si.	76
No.	13
No respondió	11
Total	100

IMAGEN 4. PREGUNTA3

### Si el conjunto tuviera un sistema Web lo usaria.



3. Si el conjunto tuviera una forma de comunicación WEB, la usaria

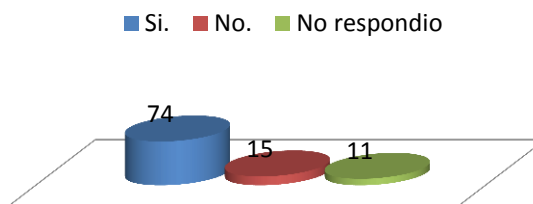
Fuente: Autores

**Imagen 4. Pregunta tres:** En la imagen se visualiza el grafico generado de la pregunta tres de la encuesta realizada a los residentes.

4. Usaría un sistema WEB para comunicarse con la administración, desde su oficina u otro lugar.	
Si.	74
No.	15
No respondió	11
Total	100

IMAGEN 5. PREGUNTA4

### Usaria un sistema Web para la gestión de la administración del conjunto.



4. Usaria un sistema WEB para comunicarse con la adminsitración, desde su oficina u otro lugar.

Fuente: Autores

**Imagen 5. Pregunta cuatro:** En la imagen se visualiza el grafico generado de la pregunta tres de la encuesta realizada a los residentes.

## **2.6. METODOLOGÍA.**

Se utilizaron 2 metodologías base para el desarrollo de este proyecto, identificadas en las fases del desarrollo de software.

Análisis y diseño:

La primera metodología utilizada Rational Unified Process (RUP) para la definición del alcance, objetivos del proyecto y definición del diseño de la aplicación software. En el cual durante el documento observaran la definición de casos de uso, diagramas de paquetes, diagramas de clases, estructura del código, cronograma, prototipos y diseño de pantallas del sistema. Esto nos permitió definir los siguientes aspectos:

Convertir los requerimientos al diseño del software.

Definición de la arquitectura de datos y de funcionamiento del sistema.

Definir el patrón de diseño (MVC Modelo Vista Control).

Desarrollo, pruebas e implementación:

Para estas fases del desarrollo de software utilizamos la metodología SCRUM, razón en la que se tomaron los requerimientos definidos como la pila de actividades que se debían realizar y se desarrollaban por grupos funcionales para tener muestras previas de funcionalidades del sistema objetivo (Sprints ó iteraciones).

Durante cada iteración evaluamos los alcances y diferentes problemas que se tenían en el desarrollo e implementación, resolviéndolo discutiendo diferentes soluciones a los problemas específicos con el director de proyecto.

SCRUM es una metodología que incrementa la rapidez y flexibilidad en el desarrollo de productos, permitiéndonos determinar las prioridades funcionales y lograr observar avances en la finalización de cada sprint.

## **3. MARCO REFERENCIAL.**

### **3.1. MARCO HISTÓRICO**

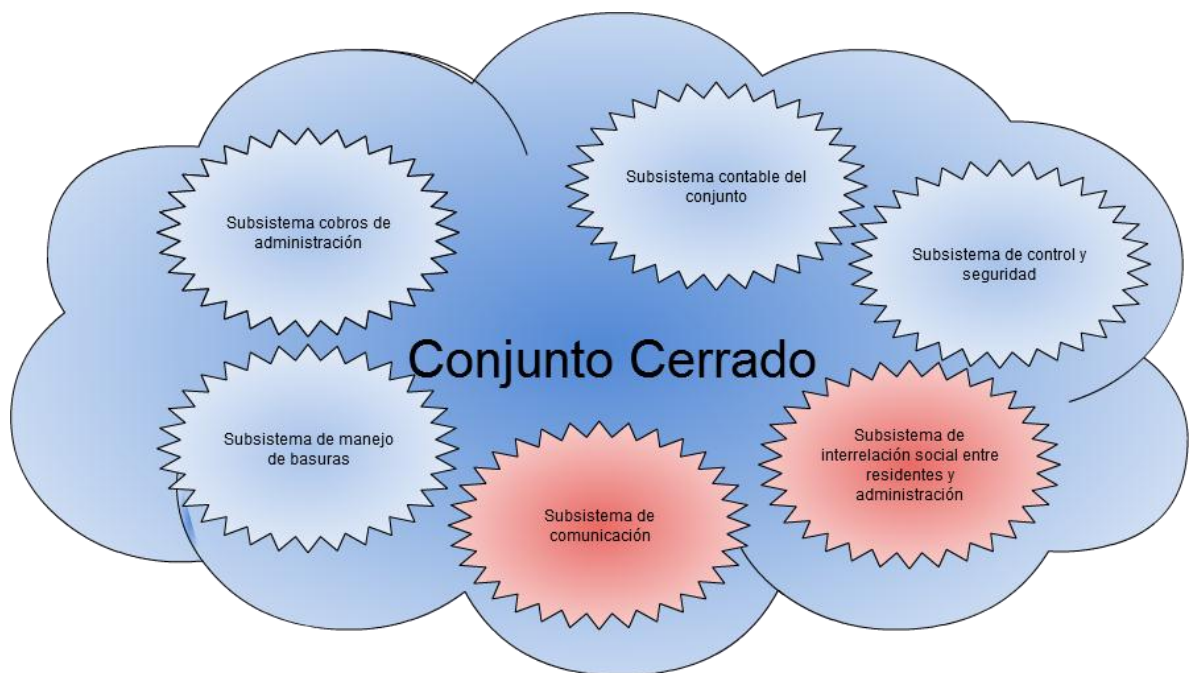
La idea de desarrollar un software para ayudar a la administración del conjunto nace en el año en agosto de 2007, en la materia de Bases de Datos, donde nos dimos cuenta que el sistema actual que manejan los conjuntos cerrados no es eficiente. Observamos que no existe la comunicación necesaria entre los residentes con la administración, por falta de tiempo o cruce de horarios en la atención; así que pensamos en proponer la idea de mejorar el sistema actual por

medio de un software que permita una mejor comunicación y gestión para mejorar los procesos que se efectúan actualmente en el conjunto.

A través de la teoría general de los sistemas, simplificamos a varios subsistemas del sistema “conjunto cerrado” tales como:

- Subsistema de cobros de administración.
- Subsistema contable del conjunto.
- Subsistema de comunicación.
- Subsistema de interrelación social entre residentes y administración.

IMAGEN 6. SUBSISTEMAS CONJUNTO CERRADO.



Fuente: Autores

**Imagen 6. Subsistemas conjunto cerrado:** En la imagen se visualiza los subsistemas que maneja el conjunto cerrado y resaltamos los dos en rojo, los cuales son en los que nos basamos para realizar el desarrollo SIACC.

De estos subsistemas decidimos realizar un enfoque de sistemas al de comunicación y de interrelación social, para dar solución a la problemática.

Siendo el año 2010 decidimos proponer la idea a la administración a tres conjuntos y a la universidad como proyecto de grado.

En noviembre del 2010 se presento la propuesta del proyecto a cargo de los estudiantes Yina Paola Chaves Silva y Miguel Ángel Colorado, bajo la supervisión

del Ingeniero Giovanni Briceño. El día 16 de febrero la propuesta es aceptada para ponerla en curso.

La TGS (La teoría general de sistemas) nos permite ser utilizado como método de investigación con el enfoque de sistemas de acuerdo con el siguiente proceso<sup>22</sup>:

1. Análisis del ambiente ó entorno.
2. Establecimiento de alcance y objetivos del sistema.
3. Definición de recursos para cumplir los objetivos.
4. Modelación del sistema, estructura organizacional.
5. Implantación del modelo.
6. Evaluación de resultados para la retroalimentación.

### **3.1.1. Diagnostico del estado actual:**

Actualmente el sistema de administración de los conjuntos encuestados maneja su administración en un horario de 8am a 5 pm y sábados de 8am a 1pm u horarios de medio tiempo de lunes a sábado. Por lo tanto los propietarios deben acercarse a la administración en esos horarios para poder pagar el valor del servicio y para consultar cualquier inquietud que se presente lo cual causa en algunos casos inconformidad porque el horario no es flexible para poder establecer una mejor comunicación. Esto se evidencio en las 100 encuestas realizadas en los dos conjuntos del sector de Cortijo y Ciudadela Colsubsidio.

El sistema de los conjuntos maneja una administración de recaudo mensual a los propietarios de cada apartamento y parqueadero. Se presta el servicio de aseo, vigilancia y préstamo del salón comunal a los propietarios.

La administración contrata a las empresas de vigilancia y de aseo para el conjunto, teniendo sus servicios activos permanentemente.

Las cuentas mensuales de administración y de parqueadero del conjunto las organiza el administrador y el revisor fiscal para presentarlos ante los representantes de los propietarios en la asamblea, dejando un registro de las actividades desarrolladas y proyectos por desarrollar en actas de cada reunión.

Existen deficiencias en el manejo y actualización de las cuentas para todas las personas que pertenecen al conjunto por varias razones:

-Ineficiencia en el flujo de la información de las cuentas de administración y de parqueadero para cada uno de los propietarios.

---

<sup>22</sup> [En línea] disponible en internet <http://www.slideshare.net/AireSum/los-mtodos-en-la-sociologia>

- No existen informes instantáneos del balance de la administración.
- Falta de comunicación entre la administración y la asamblea de propietarios para la gestión en los procesos administrativos y la toma de decisiones.
- Falta de control de la organización del personal de las empresas prestadoras de servicios al conjunto (Empresa de aseo y vigilancia).

### **3.1.2. Otros sistemas similares:**

#### **SOFTWARE DE FACTURACION Y CARTERA – (SOFCART)<sup>23</sup>**

##### **Control de Conjuntos Cerrados**

- Listado de Saldos Pendientes por Vehículos.
- Listado de Saldos por Propietarios.
- Relación de Pagos por Fecha
- Relación de Pagos Recibidos por Inmueble.
- Informe de la Cartera Vencida.
- Listado de Notas Créditos y Notas Débitos.
- Informe de Estado de los Inmuebles.

### **3.2. MARCO TEÓRICO**

#### **3.2.1. Sistemas de información.**

Los autores Laudon y Laudon (2004) definen los sistemas de información como: “Un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización. Además de apoyar a la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores a analizar problemas, a visualizar asuntos complejos y a crear productos/servicios nuevos”.

La definición proporcionada por estos autores resalta 3 aspectos básicos de los sistemas de información.

1. Un sistema de información está formado por un conjunto de componentes.
2. El segundo aspecto son las acciones que realizan son sistemas de información, las cuales son la recolección, el procesado, el almacenamiento y la distribución de la información introducida.

---

<sup>23</sup>**SOFCART** [En línea] disponible en internet URL: <http://www.puntoexe.com.co/web/es/sofcar>

3. Este aspecto hace referencia a la utilidad y a las funciones. Estas funciones son: Apoyar la toma de decisiones, el control de la organización.

### 3.2.2. Ingeniería de software:

La definición que propuso Fritz Bauer en una conferencia fundamental sobre la materia dice:<sup>24</sup>

“La ingeniería del software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales.

La Ingeniería de Software es una disciplina de la Ingeniería que concierne a todos los aspectos de la producción de software. Los Ingenieros de Software adoptan un enfoque sistemático para llevar a cabo su trabajo y utilizan las herramientas y técnicas necesarias para resolver el problema planteado, de acuerdo a las restricciones de desarrollo y recursos disponibles.”

El Proceso de Software:

Conjunto estructurado de actividades requeridas para desarrollar un sistema de software.

- Especificación-que debe hacer el software y cuáles son sus especificaciones de desarrollo.
- Desarrollo –producción del sistema de software.
- Validación–verificar que el software hace lo que el cliente pide.
- Evolución–cambiar/adaptar el software a las demandas.
- Las actividades varían dependiendo de la organización y del tipo de sistema a desarrollarse.
- Debe estar explícitamente modelado si va a ser bien administrado.

El proceso genérico ó unificado del software es:

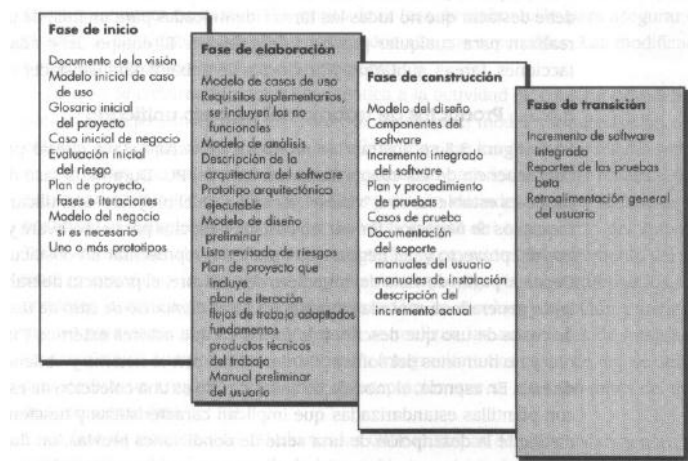
- ✓ **Especificación**-establecer los requerimientos y restricciones del sistema
- ✓ **Diseño**-Producir un modelo en papel del sistema
- ✓ **Manufactura**-construir el sistema
- ✓ **Prueba**-verificar que el sistema cumpla con las especificaciones requeridas
- ✓ **Instalación**-entregar el sistema al usuario y asegurar su operacionalidad
- ✓ **Mantenimiento**-reparar fallos en el sistema cuando sea descubiertos

### IMAGEN 7. FASES DE LA INGENIERÍA DE SOFTWARE

---

<sup>24</sup> **Ingeniería de software**, Roger S. Pressman, Ingeniería de software, un enfoque practico, Sexta edición, Mc Graw Hill.





Fuente: Autores

**Imagen 7. Fases de la ingeniería de software:** En la imagen se visualiza las cuatro fases de la ingeniería de software, explicado anteriormente.

### 3.2.3. Conceptos de desarrollo de software:

#### 3.2.3.1. Desarrollo de aplicaciones web con JEE 6<sup>25</sup>:

Según Oracle expone en su página web<sup>26</sup> el desarrollo de JEE6 para el desarrollo de aplicaciones web y actualizaciones sobre otras aplicaciones definiéndolo como:

“Java Platform, Enterprise Edition (Java EE) 6 es el estándar industrial para la empresa de computación Java. Utilizar el nuevo y ligero de Java EE6 WEB Profile para crear aplicaciones web de última generación y toda la potencia de la plataforma Java EE 6 para las aplicaciones empresariales. Los desarrolladores se beneficiarán de mejoras en la productividad con más anotaciones, más POJOs, empaque simplificado, y menos de configuración XML. ”

Por lo tanto, la nueva versión Java EE 6 nos permite ingresar a nuevas funciones y utilidades de desarrollo y facilidades de uso de recursos en combinación con un IDE especializado en Java como lo es Netbeans. También tiene soporte sobre el acceso a los datos con java mediante el api jdbc y las particularidades de las aplicaciones web en ese ámbito.

<sup>25</sup> **JEE6**, Thierry GROUSSARD, JAVA ENTERPRISE EDITION. Desarrollo de aplicaciones web con JEE 6 [en línea] disponible en internet URL: <http://www.librosaulamagna.com/libro/JAVA-ENTERPRISE-EDITION.-Desarrollo-de-aplicaciones-web-con-JEE-6/347379/8695>

<sup>26</sup> **JEE** en línea] disponible en internet URL: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>

### 3.2.2.2. Herramienta de desarrollo:

- **Netbeans:** Es un IDE de código abierto de desarrollo integrado para desarrolladores de software. Teniendo todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, así como C / C + +, PHP, JavaScript, Groovy, y Ruby.<sup>27</sup>

### 3.2.4. Bases de datos:

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. A continuación te presentamos una guía que te explicará el concepto y características de las bases de datos.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.<sup>28</sup>

#### 3.2.4.1. Definición de base de datos:

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

#### 3.2.4.2. Características

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.

---

<sup>27</sup> **NETBEANS** [en línea] disponible en internet URL: <http://netbeans.org/features/index.html>

<sup>28</sup> **BASE DE DATOS** [en línea] disponible en internet URL: <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

#### **3.2.4.3. Sistema de Gestión de Base de Datos (SGBD)**

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

#### **3.2.4.4. Ventajas de las bases de datos**

- **Control sobre la redundancia de datos:**

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

- **Consistencia de datos:**

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

- **Compartición de datos:**

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base

de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

- **Mantenimiento de estándares:**

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

- **Mejora en la integridad de datos:**

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

- **Mejora en la seguridad:**

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

- **Mejora en la accesibilidad a los datos:**

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

- **Mejora en la productividad:**

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

- **Mejora en el mantenimiento:**

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

- **Aumento de la concurrencia:**

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

- **Mejora en los servicios de copias de seguridad:**

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

#### **3.2.4.5. Desventajas de las bases de datos**

- **Complejidad:**

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

- **Coste del equipamiento adicional:**

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

- **Vulnerable a los fallos:**

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

- **Tipos de Campos**

Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

- **Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.
- **Booleanos:** poseen dos estados: Verdadero “Si” y Falso “No”.
- **Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.
- **Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- **Alfanuméricos:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
- **Autoincrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.

#### **3.2.4.6. Motores de Base de Datos**

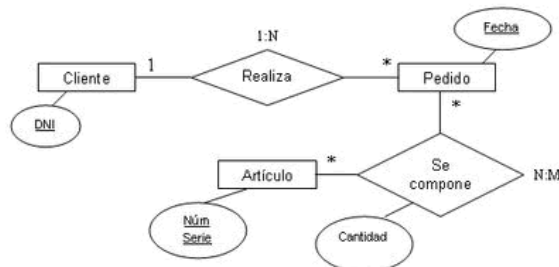
Entre los diferentes motores de base de datos, podemos encontrar los siguientes:

- MySQL: es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.
- PostgreSQL y Oracle: Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- Access: Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb con la estructura ya explicada.
- Microsoft SQL Server: es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

### 3.2.4.7. Modelo entidad-relación

Los diagramas o modelos entidad-relación (denominado por su siglas, ERD "DiagramEntityrelationship") son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

IMAGEN 8. EJEMPLO DE MER.



Fuente: Autores

**Imagen 8. Ejemplo de MER:** En la imagen se visualiza un ejemplo de cómo se debe realizar un Modelo Entidad Relación con sus correspondientes relaciones.

#### 3.2.4.7.1. Cardinalidad de las Relaciones:

El diseño de relaciones entre las tablas de una base de datos puede ser la siguiente:

- **Relaciones de uno a uno:** una instancia de la entidad A se relaciona con una y solamente una de la entidad B.

- **Relaciones de uno a muchos:** cada instancia de la entidad A se relaciona con varias instancias de la entidad B.
- **Relaciones de muchos a muchos:** cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

#### 3.2.4.8. Estructura de una Base de Datos

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos.

Por consiguiente una base de datos posee el siguiente orden jerárquico:

- Tablas
- Campos
- Registros
- Lenguaje SQL

El lenguaje SQL es el más universal en los sistemas de base de datos. Este lenguaje nos permite realizar consultas a nuestras bases de datos para mostrar, insertar, actualizar y borrar datos.

A continuación veremos un ejemplo de ellos:

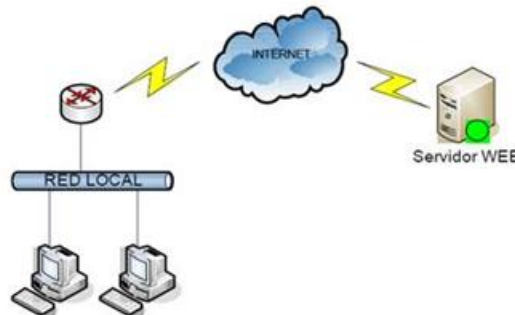
- **Mostrar:** para mostrar los registros se utiliza la instrucción Select. `Select * From comentarios.`
- **Insertar:** los registros pueden ser introducidos a partir de sentencias que emplean la instrucción Insert. `InsertInto comentarios (titulo, texto, fecha) Values ('saludos', 'como esta', '22-10-2007')`
- **Borrar:** Para borrar un registro se utiliza la instrucción Delete. En este caso debemos especificar cuál o cuáles son los registros que queremos borrar. Es por ello necesario establecer una selección que se llevara a cabo mediante la cláusula Where. `DeleteFrom comentarios Where id='1'.`
- **Actualizar:** para actualizar los registros se utiliza la instrucción Update. Como para el caso de Delete, necesitamos especificar por medio de Where cuáles son los registros en los que queremos hacer efectivas nuestras modificaciones. Además, tendremos que especificar cuáles son los nuevos valores de los campos que deseamos actualizar. `Update comentarios Set titulo='Mi Primer Comentario' Where id='1'.`



### 3.2.5. Servidores WEB<sup>29</sup>:

Es el programa que, utilizando el protocolo de comunicaciones HTTP, es capaz de recibir peticiones de información de un programa cliente (navegador), recuperar la información solicitada y enviarla al programa cliente para su visualización por el usuario a través de internet.

IMAGEN 9. DIAGRAMA DE UN SERVIDOR WEB



Fuente: Autores

**Imagen 9. Diagrama de un servidor Web:** En la imagen se visualiza un ejemplo de diagrama de servidor WEB, mostrando el concepto teórico de Servidores Web.

En el desarrollo de del proyecto utilizaremos como servidor web Apache Tomcat *Apache Tomcat*<sup>30</sup> es una implementación de software de código abierto de las tecnologías Java Servlet y JavaServerPages. El Java Servlet y páginas JavaServer son especificaciones que se desarrollan bajo la Java CommunityProcess.

Apache Tomcat es un entorno de desarrollado abierto y participativo y publicado bajo la licencia Apache versión 2. Apache Tomcat es la intención de ser una colaboración de los de su clase en ser de los mejores desarrollados de todo el mundo.

Apache Tomcat posee numerosas competencias a gran escala, aplicaciones web de misión crítica a través de una amplia gama de industrias y organizaciones.

### 3.2.6. Metodología RUP.

Las siglas RUP en ingles significa RationalUnifiedProcess (Proceso Unificado de Rational) es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidad es dentro de una

<sup>29</sup>**SERVIDORES WEB** [en línea] disponible en internet URL:

<http://www.aranzadi.es/index.php/informacion-juridica/informacion-interes/glosario-de-terminos-sobre-internet-y-spam>

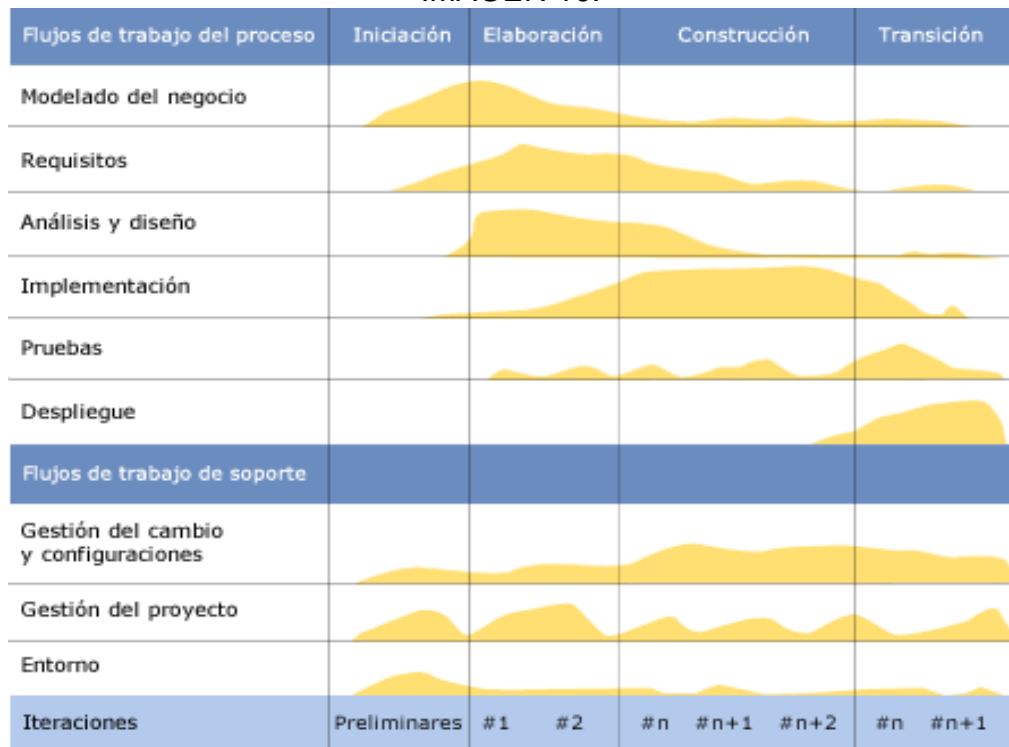
<sup>30</sup>**APACHE TOMCAT** [en línea] disponible en internet URL: <http://tomcat.apache.org/index.html>

organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos.<sup>31</sup>

### 3.2.6.1. Ciclo de vida de RUP

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades.

IMAGEN 10.



CICLO DE VIDA DE RUP

Fuente: Autores

**Imagen 10. Ciclo de vida de RUP:** En la imagen se visualiza el modelo de división de fases de la metodología RUP.

*En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades*

<sup>31</sup> **RUP** [en línea] disponible en internet URL:  
[http://webcache.googleusercontent.com/search?q=cache:G00C7bQVAt0J:biblioteca.usac.edu.gt/tesis/08/08\\_7691.pdf+metodologia+RUP&hl=es&ct=clnk&cd=6&gl=co&client=firefox-a](http://webcache.googleusercontent.com/search?q=cache:G00C7bQVAt0J:biblioteca.usac.edu.gt/tesis/08/08_7691.pdf+metodologia+RUP&hl=es&ct=clnk&cd=6&gl=co&client=firefox-a)

- Inicio: Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos. Se define el alcance del proyecto.
- Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- Transición: se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.<sup>32</sup>

### **3.2.7. SCRUM**

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

#### **3.2.7.1 El proceso de SCRUM**

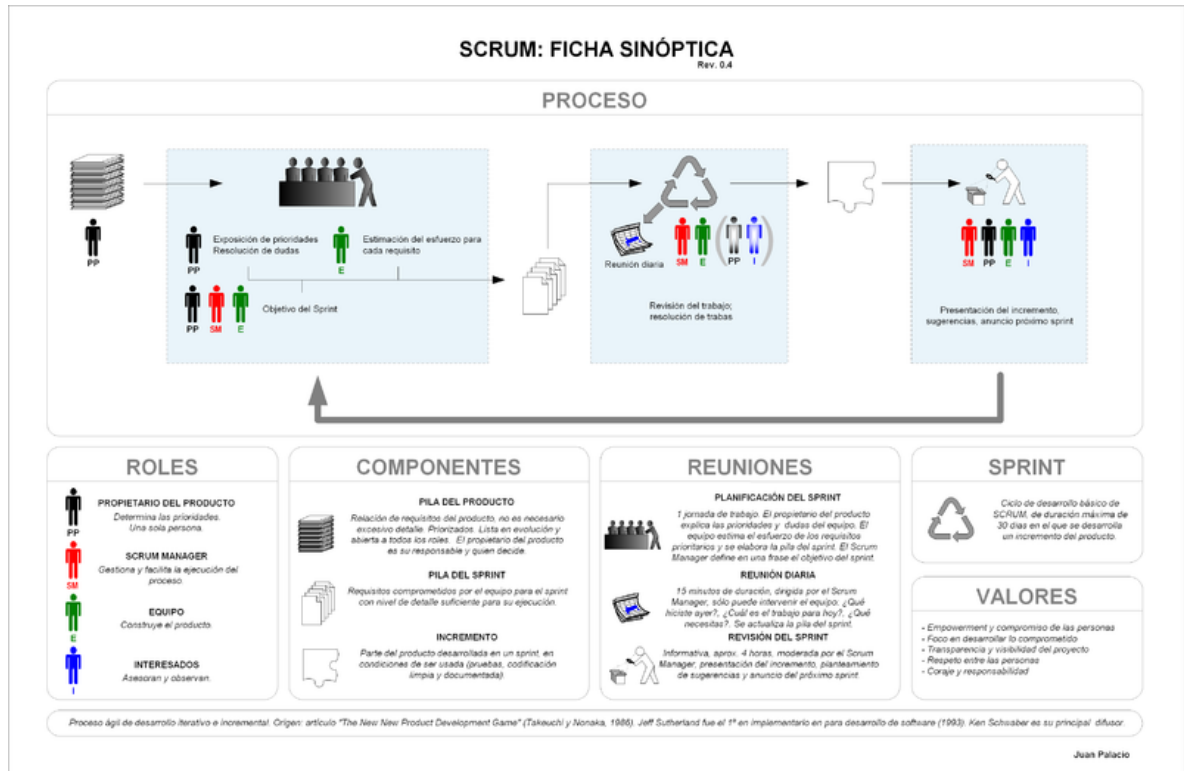
En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que

---

<sup>32</sup>**CICLO DE VIDA DE RUP** [en línea] disponible en internet URL:  
<http://es.scribd.com/doc/297224/RUP>

sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

IMAGEN 11. IMAGEN SCRUM



<sup>33</sup> Fuente: Autores

**Imagen 11.Imagen SCRUM:** En la imagen se visualiza la estructura de la metodología de desarrollo Scrum, la cual se usó en SIACC.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la re planificación de objetivos que realiza al inicio de cada iteración.

Las actividades que se llevan a cabo en Scrum son las siguientes:

<sup>33</sup> **IMAGEN SCRUM** [en línea] disponible en internet URL:  
[http://es.wikipedia.org/wiki/Archivo:Ficha\\_scrum.png](http://es.wikipedia.org/wiki/Archivo:Ficha_scrum.png)

### 3.2.7.2. Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. Selección de requisitos (4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
2. Planificación de la iteración (4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.<sup>34</sup>

## 3.3. MARCO CONCEPTUAL

### 3.3.1. Frameworks

Plantillas predefinidas, que facilitan la programación, como Eclipse o Netbeans.

En el desarrollo de *software*, un *framework* o *infraestructura digital*, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de *software* concretos, con base a la cual otro proyecto de *software* puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.<sup>35</sup>

### 3.3.2. JUDE (Java and UML Developers Environment)

Ambiente para Desarrolladores UML y Java es un software para el modelamiento en UML. Esta desarrollado en Java, lo que garantiza su portabilidad para cualquier plataforma que posee una máquina virtual Java.

---

<sup>34</sup> **SCRUM** [en línea] disponible en internet URL: <http://www.proyectosagiles.org/que-es-scrum>

<sup>35</sup> **FRAMEWORKS** [en línea] disponible en internet URL: <http://es.wikipedia.org/wiki/Framework>

JUDE es una plataforma que desarrolla dos aplicaciones para el modelado utilizando el lenguaje UML. La primera de ellas es JUDE Professional, una herramienta de pago con la que se pueden realizar los siguientes tipos de diagramas: UML, Entidad-Relacion, DFD, Flowchart y CRUD. En segundo lugar está la aplicación JUDE Community, herramienta libre para el modelado a través del uso de lenguaje UMLy cuya última versión soporta parcialmente la especificación 2.0 de dicho lenguaje. Me centraré en esta última, ya que la reseña que se nos hacía en la asignatura era sobre la herramienta libre.

Además de soportar parcialmente la versión 2.0 de UML, esta herramienta también acepta diagramas pertenecientes a la versión 1.4 del mismo. Además, permite al usuario importar y exportar ficheros escritos en el lenguaje Java y ofrece soporte para la creación automática de diagramas de clases. Permite exportar el trabajo realizado en documentos HTML y también ofrece la posibilidad de exportar los diagramas realizados a ficheros del tipo PNG y JPEG.<sup>36</sup>

### 3.3.3. IDE Netbeans

Netbeans se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma Netbeans.

Netbeans es un proyecto de Software Libre. SunMicroSystems fundó el proyecto de código abierto Netbeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma Netbeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las Apis de Netbeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El IDE NetBeans es un IDE - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número

---

<sup>36</sup> **JUDE** [en línea] disponible en internet URL:

[http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CH0QFjAB&url=http%3A%2F%2Fpetra.euitio.uniovi.es%2F~i1667065%2FHD%2Fdocumentos%2FHerramientasOO.doc&ei=PggfUNK\\_DeLY0QGrzYGwDg&usg=AFQjCNH-RytiXDvf\\_\\_IlbLaDRzYKdxGoYQ&sig2=b610gPOjDOyX1929xmLH7Q](http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CH0QFjAB&url=http%3A%2F%2Fpetra.euitio.uniovi.es%2F~i1667065%2FHD%2Fdocumentos%2FHerramientasOO.doc&ei=PggfUNK_DeLY0QGrzYGwDg&usg=AFQjCNH-RytiXDvf__IlbLaDRzYKdxGoYQ&sig2=b610gPOjDOyX1929xmLH7Q)

importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto de Software Libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Apache Ant, control de versiones y refactoring.<sup>37</sup>

### **3.3.4. Stakeholder**

Personas u organizaciones son participantes funcionales directos en el proyecto y en el desarrollo. Con base en ellos se toman decisiones sobre la funcionalidad y características del producto de software.

### **3.3.5. Tomcat**

También llamado JakartaTomcat o Apache Tomcat, funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServerPages (JSP) de Sun Microsystems.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.<sup>38</sup>

---

<sup>37</sup> **NETBEANS** [en línea] disponible en internet URL: <http://www.dosideas.com/wiki/NetBeans>

<sup>38</sup> **TOMCAT** [en línea] disponible en internet URL: <http://www.raulrodriguezcabestany.com/apache-tomcat>

### 3.3.6. UML

El Lenguaje de Modelado Unificado (UML:UnifiedModelingLanguage) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s.UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos de un lenguaje de modelado y de un proceso. El UML , fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE (Booch, G. et al., 1999). UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios.

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo.<sup>39</sup>

### 3.3.7. Workflow

Se refiere al flujo de trabajo a seguir para la consecución de una tarea o trabajo predeterminado. Se define como un sistema de secuencia de tareas de un proceso de negocio. Su definición y control puede ser manual, informatizado o mixto. Organiza y controla tareas, recursos y reglas necesarias para completar el proceso de negocio.<sup>40</sup>

### 3.3.8. XML (eXtensibleMarkupLanguage)

Significa eXtensiblemarkuplanguage, o lenguaje de anotación extensible. Ya conocemos el lenguaje HTML (hypertextmarkuplanguage), lenguaje de anotación para página webs que permite navegación tipo hipertexto; sin embargo, XML no es sólo un lenguaje, es una forma de especificar lenguajes, de ahí lo de extensible.

---

<sup>39</sup> **UML** [en línea] disponible en internet URL: <http://www.docirs.cl/uml.htm>

<sup>40</sup> **WORKFLOW** [en línea] disponible en internet URL: <http://www.pixelware.com/workflow-flujo-trabajo.htm>



XML se inició como un subconjunto de SGML (structuredgeneralizedmarkuplanguage), un standard ISO para documentos estructurados que es sumamente complejo para poder servir documentos en la web. XML es algo así como SGML simplificado, de forma que una aplicación no necesita comprender SGML completo para interpretar un documento, sino sólo el subconjunto que se defina. Los editores SGML, sin embargo, pueden comprender XML.<sup>41</sup>

### 3.4 MARCO LEGAL

#### 3.4.1. La ley 675 de 2001, de la propiedad horizontal cita:

ARTÍCULO 1o. OBJETO. La presente ley regula la forma especial de dominio, denominada propiedad horizontal, en la que concurren derechos de propiedad exclusiva sobre bienes privados y derechos de copropiedad sobre el terreno y los demás bienes comunes, con el fin de garantizar la seguridad y la convivencia pacífica en los inmuebles sometidos a ella, así como la función social de la propiedad.

ARTÍCULO 3o. DEFINICIONES.

“Conjunto: Desarrollo inmobiliario conformado por varios edificios levantados sobre uno o varios lotes de terreno, que comparten, áreas y servicios de uso y utilidad general, como vías internas, estacionamientos, zonas verdes, muros de cerramiento, porterías, entre otros.”

La inscripción de un edificio o conjunto, es la inscripción de las personas jurídicas sometidas al régimen de propiedad horizontal (edificios y conjuntos) en el registro de propiedad horizontal que lleva el IDUC, conforme a lo establecido en la Ley 675 de 2001.

La inscripción faculta a los edificios y conjuntos para ejercer derechos y obligaciones. Es un documento indispensable para la obtención del NIT, abrir una cuenta bancaria, otorgar poder a un abogado para iniciar procesos judiciales de cobro de la cartera morosa, afiliar a los empleados a la seguridad social, etc.

ARTÍCULO 4o. CONSTITUCIÓN. Un edificio o conjunto se somete al régimen de propiedad horizontal mediante escritura pública registrada en la Oficina de Registro de Instrumentos Públicos. Realizada esta inscripción, surge la persona jurídica a que se refiere esta ley.

---

<sup>41</sup>XML [en línea] disponible en internet URL: <http://geneura.ugr.es/~jmerelo/xml/>

ARTÍCULO 5o. CONTENIDO DE LA ESCRITURA O REGLAMENTO DE PROPIEDAD HORIZONTAL. La escritura pública que contiene el reglamento de propiedad horizontal deberá incluir como mínimo, los siguientes temas:

1. El nombre e identificación del propietario.
2. El nombre distintivo del edificio o conjunto.
3. La determinación del terreno o terrenos sobre los cuales se levanta el edificio o conjunto, indicando detalladamente su nomenclatura, área y linderos, así como el título o títulos de adquisición y los correspondientes folios de matrícula inmobiliaria.
4. La identificación de cada uno de los bienes de dominio particular de acuerdo con los planos aprobados por cualquiera de las curadurías urbanas de Bogotá.
5. La determinación de los bienes comunes, con indicación de los que tengan el carácter de esenciales, no esenciales y los de uso exclusivo.
6. Los coeficientes de copropiedad.
7. Los módulos de contribución, sólo para los edificios o conjuntos de carácter comercial y mixto.
8. La destinación de los bienes de dominio particular que conforman el edificio o conjunto, es decir, explicar que son: vivienda, oficina, locales comerciales, industriales o mixtos.
9. Las especificaciones de construcción y condiciones de seguridad y salubridad del edificio o conjunto.
10. La participación de los propietarios en las expensas comunes.
11. La constitución un fondo de imprevistos para atender obligaciones o expensas imprevistas, el cual se formará e incrementará con un porcentaje de recargo de mínimo el uno por ciento (1%) sobre el presupuesto anual de los gastos comunes.
12. Establecer los derechos, deberes y prohibiciones de los propietarios de bienes privados con respecto al edificio o conjunto.
13. Determinar los órganos de administración, dirección y control, así como las reglas que gobiernan la organización y funcionamiento del edificio o conjunto. Este punto se refiere a la determinación de funciones, convocatoria, clases de reuniones, quórum, actas, etc. de la asamblea y el administrador, al igual que el

Consejo de Administración en los edificios y conjunto que sea necesario u obligatorio.

14. El reglamento de administración de la propiedad horizontal, determinándose las funciones del administrador, responsabilidades y faltas del administrador.

15. La creación del comité de convivencia.

16. Las causales y procedimiento para la extinción de la propiedad horizontal y la liquidación de la personería jurídica.

17. Anexar la licencia de construcción y los planos aprobado por la respectiva curaduría urbana que muestren la localización, linderos, nomenclatura y área de cada una de las unidades privadas y las áreas y los bienes de uso común \*.

Sólo en los conjuntos integrados por etapas, la escritura de constitución deberá indicar esta circunstancia, y regular dentro de su contenido el régimen general del mismo, la forma de integrar las etapas subsiguientes, y los coeficientes de copropiedad de los bienes privados de la etapa que se conforma, los cuales tendrán carácter provisional.

Las subsiguientes etapas las integrará el propietario inicial, mediante escrituras adicionales, en las cuales se identificarán sus bienes privados, los bienes comunes localizados en cada etapa y el nuevo cálculo de los coeficientes de copropiedad de la totalidad de los bienes privados de las etapas integradas al conjunto, los cuales tendrán carácter provisional.

En la escritura pública por medio de la cual se integra la última etapa, los coeficientes de copropiedad de todo el conjunto se determinarán con carácter definitivo.

\* Si el notario público encuentra una o más discordancia entre la escritura de constitución o adición al régimen de propiedad horizontal y la licencia y los planos deberá dejar constancia expresa en la escritura.

\*\*Se recomienda que la elaboración del reglamento de propiedad horizontal se le encomiende a un abogado experto en el tema, quien debe contar con el apoyo de un arquitecto.

### 3.5. MARCO TECNOLÓGICO.

SIACC es un sistema el cual maneja para su ejecución y utilización la siguiente plataforma tecnológica:

TABLA1: MARCO TECNOLÓGICO.	
COMPONENTE	DESCRIPCIÓN
Sistema Operativo	Multiplataforma
Lenguaje de Programación	JAVA
Base de Datos	Postgres 8.4
Ambiente de Desarrollo	Netbeans 7.1
Servidor Web	Apache Tomcat 6.18 o superior
Lenguaje de Modelado	UML
Navegadores	Chrome
Biblioteca de Persistencia	Eclipse link JPA 2.0
Librería de componente visual	Richfaces 4.2

#### 3.5.1. Recursos Hardware para el desarrollo del proyecto:

Los recursos de hardware (los 2 equipos de desarrollo) han sido comprados legalmente y los recursos de software con su licencia.

#### 3.5.2. Recursos Software para el desarrollo del proyecto:

❖ **Sistema operativo:**

El sistema se desarrollo en Windows 7 Home Premium y Basic. Con licencias originales de Microsoft para el desarrollo de los portátiles descritos en la factibilidad económica y técnica.

❖ **Motor de base de datos: Postgres 8.4:**<sup>42</sup>

---

<sup>42</sup>**POSTGRES** [en línea] disponible en internet URL:  
<http://www.postgresql.org/about/press/presskit83.html.es>

PostgreSQL se distribuye bajo licencia BSD, lo cual permite el uso y distribución sin costo, tanto para aplicaciones comerciales como no comerciales.

❖ **Servidor Apache Tomcat:**

Licencia de Apache es licencia del libre-software sido autor por Fundación del software de Apache (ASF).

❖ **Lenguaje Java:**

Bajo especificación JEE6, para la implementación de JSF en la aplicación.

<sup>43</sup>El cual tiene licencia GPL. Pero para poder implementar la aplicación en funcionamiento comercial como ley en Colombia es necesario tener tarjeta profesional y registrarlo en derechos de autor y en Cámara y Comercio por parte de la organización.

### 3.5.3. Ambiente de instalación:

- **Sistema operativo:** Versión para el uso todo sistema operativo que tenga instalado JAVA, Apache tomcat y Postgres 8.4.
- **Motor de base de datos: Postgres 8.4** Versión para el uso 8.4 únicamente.
- **Servidor Apache Tomcat:** Versión el uso 6.18 en adelante.
- **Netbeans versión 7.1:** Se distribuye a través de la licencia CommonDevelopment and DistributionLicense (Licencia Común de Desarrollo y Distribución) o CDDL es una licencia de código abierto.
- **Lenguaje Java:** Se necesita tener instalado el JDK 1.5 en adelante.

---

<sup>43</sup> **JAVA**[en línea] disponible en internet URL:<http://www.todoexpertos.com/categorias/tecnologia-e-internet/programacion/java/respuestas/1737936/licencia-java>

## **4. ESTRUCTURA TEMÁTICA – DESARROLLO DEL PROYECTO**

Las metodologías que se aplicaron en el desarrollo de este proyecto son RUP y Scrum, ya que se abordaron y aplicaron aspectos importantes de la ingeniería de software y desarrollo ágil, las etapas tenidas en cuenta fueron Análisis, Diseño, Desarrollo, pruebas e implementación.

### **4.1. CAPITULO 1: ANÁLISIS**

#### **4.1.1. Descripción general del sistema.**

A través de la investigación de los sistemas que interactúan dentro del conjunto cerrado, logramos definir los diferentes elementos que interactúan para obtener beneficios mutuos de convivencia. Dentro de los sistemas planteados en el marco referencial – histórico,

Teniendo en cuenta la falta de comunicación que observamos en los conjuntos cerrados, entre la administración y los propietarios, decidimos realizar una aplicación Web que mejore los vínculos entre los dos entes.

El sistema principalmente sirve para mantener informado al propietario de las cosas que pasan a diario en el conjunto (noticias), para hacer llamados a reuniones de asamblea, extraordinarias o eventos sociales (eventos), para tener una comunicación desde cualquier parte fuera del conjunto (correo), prestamos de salón comunal on-line, consultas de información administrativa por parte del propietario (informes), control de los propietarios y residentes de los apartamentos, control de las empresas que prestan servicios de aseo y vigilancia en el conjunto.

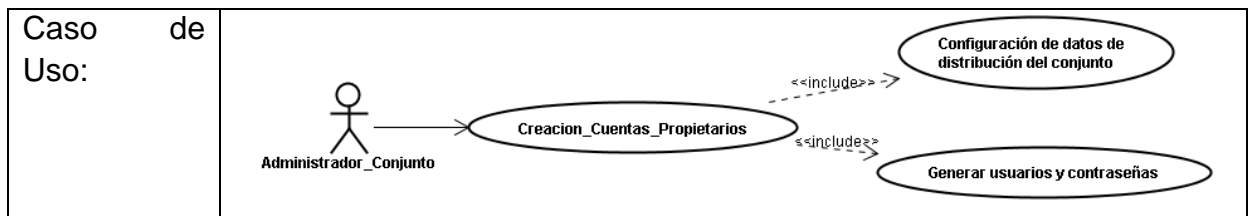
#### **4.1.2. Requerimientos Funcionales.**

Nuestro desarrollo de software lo vamos a manejar con la metodología de espiral. La cual nos brinda flexibilidad a la hora de establecer los requerimientos, teniendo en cuenta las necesidades del cliente.

Número:	R_01
Requisito:	Acceder al Sistema
Actor:	El administrador del sistema, el administrador del Conjunto y los residentes del conjunto.
Descripción:	El software administrador de conjuntos cerrados brindará acceso a

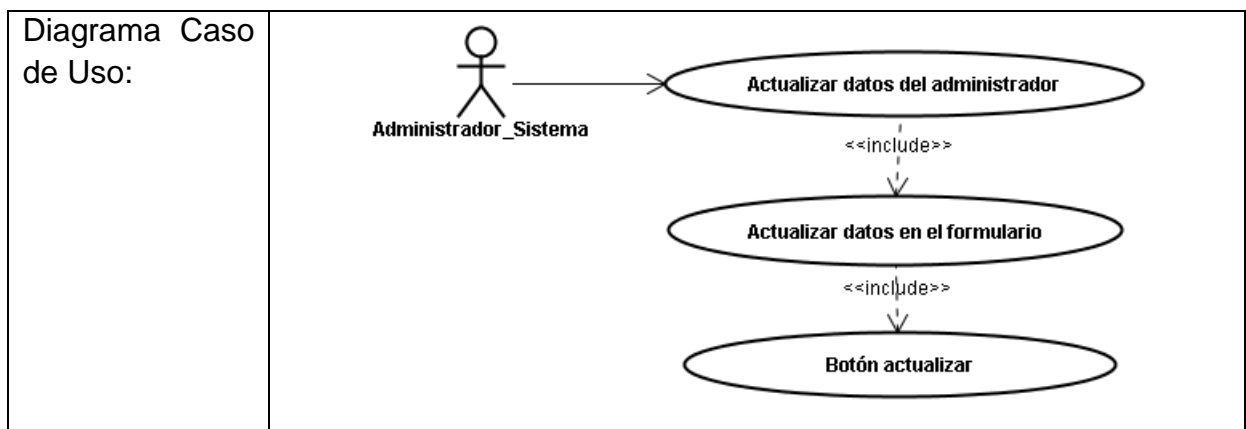
	éste ingresando los datos: identificación de usuario y su respectiva contraseña, creando una sesión de trabajo. El usuario al terminar de trabajar en el sistema, tendrá que cerrar su sesión por seguridad.
Proceso:	Para el ingreso al sistema, es necesario que el usuario se encuentre registrado en el mismo; por esta razón al momento que el usuario ingrese sus datos, el sistema valida la información y genera la respectiva información correspondiente según el tipo de usuario. En caso contrario, se informa al usuario que no se encuentra registrado en el sistema.
Entrada:	Número de Identificación (Numeric), Contraseña (Text).
Salida:	Interfaz con la información correspondiente según el perfil, además de los módulos que puede administrar.
Diagrama Caso de Uso:	<pre> graph LR     Admin[Administrador_Conjunto] --&gt; Acceder_Sistema((Acceder_Sistema))     Residentes[Residentes] --&gt; Acceder_Sistema     Acceder_Sistema -.-&gt; &lt;&lt;include&gt;&gt;  Ingresar_usuario_y_contraseña([Ingresar usuario y contraseña])     Acceder_Sistema -.-&gt; &lt;&lt;include&gt;&gt;  Validar_datos_de_registros([Validar datos de registros])     Ingresar_usuario_y_contraseña -.-&gt; &lt;&lt;include&gt;&gt;  Ingresar_usuario_al_sistema([Ingresar usuario al sistema]) </pre> <p>The diagram shows two actors, 'Administrador_Conjunto' and 'Residentes', both pointing to a central use case 'Acceder_Sistema'. From 'Acceder_Sistema', two dashed arrows labeled '&lt;&lt;include&gt;&gt;' point to 'Ingresar usuario y contraseña' and 'Validar datos de registros'. A third dashed arrow labeled '&lt;&lt;include&gt;&gt;' points from 'Ingresar usuario y contraseña' to 'Ingresar usuario al sistema'.</p>

Número:	R_02
Requisito:	Creación de cuentas de propietarios
Actor:	Administrador del conjunto
Descripción:	El usuario administrador del conjunto podrá registrar el número de apartamentos con sus respectivas cuentas.
Proceso:	El usuario administrador del conjunto proceder a ingresar el número de torres, apartamentos y secuencia personalizable, el sistema generara las cuentas del número de apartamentos ingresados anteriormente, donde él usuario tendrá: Numero de torre apartamento y la contraseña igualmente.
Entrada:	Numero de torres (Numeric), Número de apartamentos por piso (Numeric), Numero de pisos por torre (Numeric). Id_Torre (Numeric), Id_Apartamento (Numeric)
Salida:	Interfaz donde se muestre la información de las cuentas por apartamento.
Diagrama	

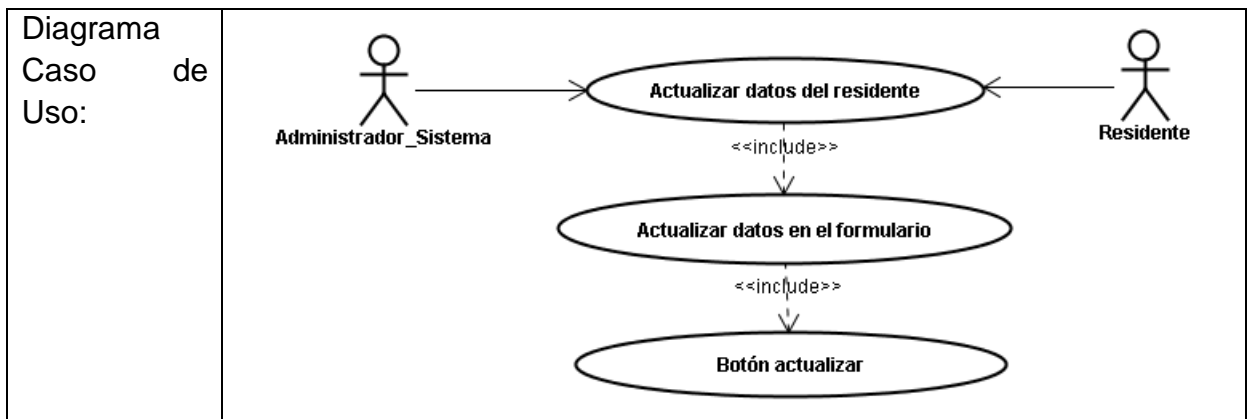


Número:	R_3
Requisito:	Actualizar datos del administrador de conjunto
Actor:	Administrador de conjunto
Descripción:	El perfil administrador del conjunto podrán actualizar sus datos: Identificación (Numeric) Nombres_Arrendatario (Text) Apellidos_Arrendatario (Text) Telefono1 (Numeric) Telefono2 (Numeric) Celular (Numeric) Correo electrónico (Text) Clave (Text)
Proceso:	El administrador del conjunto para proceder a actualizar la información, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo Perfil, en donde se realizará la respectiva modificación de los datos que han sido ingresados, según el criterio del usuario. El sistema guardará las modificaciones realizadas, actualizando los datos modificados.
Entrada:	Identificación (Numeric), Nombres_Adminitrador (Text), Apellidos_Administrador (Text), Telefono1 (Numeric), Telefono2 (Numeric), Celular (Numeric), correo electrónico(Text) y Clave (Text)
Salida:	Interfaz donde se muestre la información del usuario actualizado.





Número:	R_4
Requisito:	Actualizar datos propietarios
Actor:	Propietarios y administrador de conjunto.
Descripción:	<p>El perfil Propietario y administrador podrán actualizar los datos del propietario como lo son:</p> <p>Identificación (Numeric)</p> <p>Nombres_Propietario (Text)</p> <p>Apellidos_Propietario (Text)</p> <p>Telefono1 (Numeric)</p> <p>Telefono2 (Numeric)</p> <p>Celular (Numeric)</p> <p>Correo electrónico (Text)</p> <p>Clave (Text)</p>
Proceso:	<p>El Propietario o administrador del conjunto desde su perfil puede actualizar la información, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo Actualizar Datos del propietario, en donde se realizará la respectiva modificación de los datos que han sido ingresados, según el criterio del usuario. El sistema guardará las modificaciones realizadas, actualizando los datos modificados.</p>
Entrada:	Identificación (Numeric), Nombres_Propietario (Text), Apellidos_Propietario (Text), Telefono1 (Numeric), Telefono2 (Numeric), Celular (Numeric), correo electrónico(Text) y Clave (Text)
Salida:	Interfaz donde se muestre la información del usuario, los datos que han sido registrados, incluyendo las modificaciones realizadas.

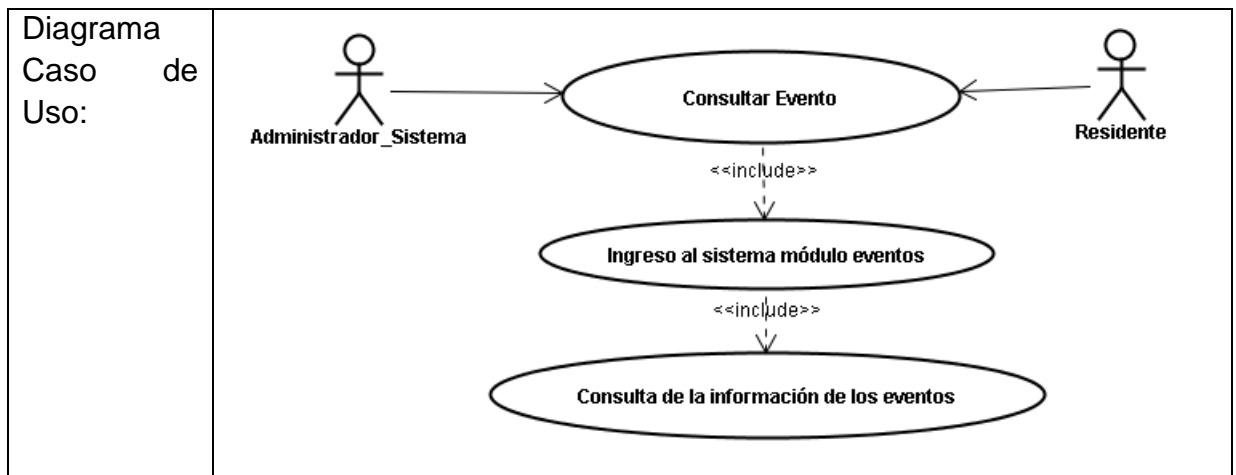


Número:	R_5
Requisito:	Consulta noticias
Actor:	Residentes y administrador del sistema
Descripción:	Los perfiles residente y administrador del sistema podrán acceder al módulo de noticias que es alimentada por el administrador.
Proceso:	Los residentes y el administrador pueden proceder a la consulta información de noticias, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar a los módulos de Noticias, en donde se realizará las respectivas consultas. El sistema le permitirá navegar mostrando la información alimentada por el administrador.
Entrada:	Identificador_Noticia (Numeric) Nombre_Noticia (Text) Contenido_Noticia (Text)
Salida:	Interfaz donde se muestre la información de la noticia.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Sistema] --&gt; UC1(Consultar Noticias)     Residente[Residente] --&gt; UC1     UC1 --&gt; &lt;&lt;include&gt;&gt;  UC2(Ingreso al sistema módulo noticias)     UC2 --&gt; &lt;&lt;include&gt;&gt;  UC3(Consulta de la información de noticias)           </pre> <p>The diagram shows two actors, 'Administrador_Sistema' and 'Residente', both connected to the use case 'Consultar Noticias'. This use case includes 'Ingreso al sistema módulo noticias', which in turn includes 'Consulta de la información de noticias'.</p>

Número:	R_6
Requisito:	Editar noticias

Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, puede modificar noticias que hayan vivido en el conjunto y cambiar nombre y texto cuerpo de la noticia.
Proceso:	El usuario administrador del conjunto, debe haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de noticias donde puede modificar la noticia. Para ello se deben modificar algunos de los campos de noticia.
Entrada:	Nombre_noticia (text) Cuerpo_noticia (text)
Salida:	Interfaz donde se muestre la información de la noticia modificada.
Diagrama Caso de Uso:	<pre> graph LR     Actor[Administrador_Conjunto] --&gt; UC1((Actualizar Noticias))     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2((Ingresar datos de la noticia))     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3((Boton actualizar))     UC3 -.-&gt; &lt;&lt;include&gt;&gt;  UC4((Publicar Noticia actualizada))           </pre> <p>The diagram shows a stick figure actor labeled 'Administrador_Conjunto' with an arrow pointing to an oval use case 'Actualizar Noticias'. From 'Actualizar Noticias', a dashed arrow labeled '&lt;&lt;include&gt;&gt;' points to an oval use case 'Ingresar datos de la noticia'. From 'Ingresar datos de la noticia', a solid arrow labeled '&lt;&lt;include&gt;&gt;' points down to an oval use case 'Boton actualizar'. From 'Boton actualizar', a dashed arrow labeled '&lt;&lt;include&gt;&gt;' points down to an oval use case 'Publicar Noticia actualizada'.</p>

Número:	R_7
Requisito:	Consulta eventos
Actor:	Residentes y administrador del sistema
Descripción:	Los perfiles residente y administrador del sistema podrán acceder al módulo de eventos que es alimentada por el administrador.
Proceso:	Los residentes y el administrador pueden proceder a la consulta información de eventos, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar a los módulos de Eventos, en donde se realizará las respectivas consultas. El sistema le permitirá navegar mostrando la información alimentada por el administrador.
Entrada:	Identificador_Evento (Numeric) Nombre_Evento (Text) Contenido_evento (Text) Fecha_Evento (Date)
Salida:	Interfaz donde se muestre la información del evento.



Número:	R_8
Requisito:	Editar eventos
Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, puede editar eventos que haya creado para el conjunto cambiar nombre y cuerpo del evento.
Proceso:	El usuario administrador del conjunto, debe haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de eventos donde puede modificar el evento. Para ello se deben modificar algunos de los campos del evento.
Entrada:	Nombre_evento (text) Nombre_evento(text) Fecha_evento publicación(Date)
Salida:	Interfaz donde se muestre la información del evento modificado.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1((Actualizar Eventos))     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2((Ingresar datos del evento))     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3((Boton actualizar))     UC3 -.-&gt; &lt;&lt;include&gt;&gt;  UC4((Publicar Evento actualizado))           </pre> <p>The diagram shows an actor 'Administrador_Conjunto' associated with the use case 'Actualizar Eventos'. This use case includes 'Ingresar datos del evento', which includes 'Boton actualizar', which finally includes 'Publicar Evento actualizado'.</p>

Número:	R_9
Requisito:	Envío de correos a residentes
Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, tendrán una opción de

	envío de correo el cual debe ser de Gmail, es el que cada cuenta tiene anexa.
Proceso:	El usuario administrador del conjunto, debe haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de Correo donde puede seleccionar torre y apartamento y escribir el asunto y cuerpo del correo electrónico, para luego poderlo enviar, solicitando su contraseña.
Entrada:	Torre (Numeric) Apartamento (Numeric) Asunto (Text) Mensaje(Text) Remitente (Text) Contraseña (Text)
Salida:	Interfaz donde se muestre que el mensaje ha sido enviado al residente.
Diagrama Caso de Uso:	<pre> graph TD     Actor[Administrador_Sistema] --&gt; UC1([Envío de correos residentes])     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2([Ingresar datos del residente])     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3([Se valida el correo del residente en la base de datos])     UC3 -.-&gt; &lt;&lt;include&gt;&gt;  UC4([Ingresar contraseña])     UC4 -.-&gt; &lt;&lt;include&gt;&gt;  UC5([Botón para enviar mensaje]) </pre>

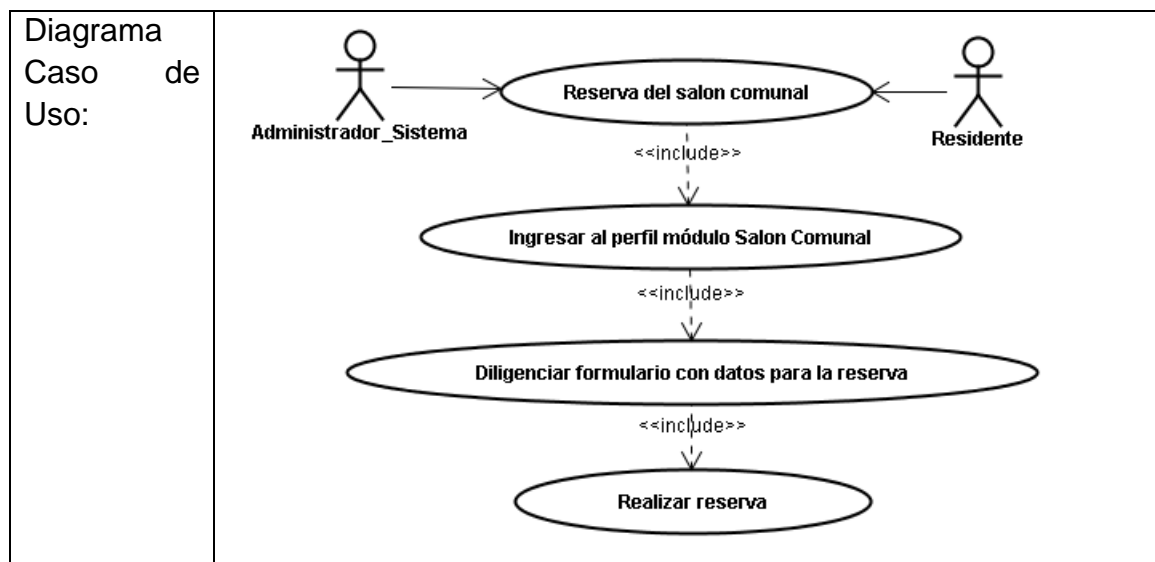
Número:	R_10
Requisito:	Envío de correos a administrador
Actor:	Residentes.
Descripción:	El perfil de residentes, tendrán una opción de envío de correo el cual debe ser de Gmail, es el que cada cuenta tiene anexa.
Proceso:	El usuario administrador del conjunto ó el residente, debe

	haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de Correo donde puede seleccionar torre y apartamento y escribir el asunto y cuerpo del correo electrónico, para luego poderlo enviar, solicitando su contraseña.
Entrada:	Torre (Numeric) Apartamento (Numeric) Asunto (Text) Mensaje(Text) Remitente (Text) Contraseña (Text)
Salida:	Interfaz donde se muestre que el mensaje ha sido enviado a su destinatario.
Diagrama Caso de Uso:	<pre> graph TD     Residente((Residente)) --&gt; UC1([Envío de correos administrador])     UC1 --&gt; &lt;&lt;include&gt;&gt;  UC2([Ingresar datos del mensaje])     UC2 --&gt; &lt;&lt;include&gt;&gt;  UC3([Consulta del correo de alministrador])     UC3 --&gt; &lt;&lt;include&gt;&gt;  UC4([Ingresar_Contraseña])     UC4 --&gt; &lt;&lt;include&gt;&gt;  UC5([Botón para enviar mensaje]) </pre>

Número:	R_11
Requisito:	Consulta de salón comunal
Actor:	Administrador del conjunto y residentes
Descripción:	El administrador y el residente pueden consultar las fechas reservadas del salón comunal.
Proceso:	Los residentes y el administrador pueden proceder a la consulta información de las fechas reservadas para el uso del salón comunal deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar a los módulos de Salón comunal,

	en donde se realizará las respectivas consultas de las reservas. El sistema le permitirá navegar mostrando la información alimentada por los usuarios.
Entrada:	Número_evento (Numeric), Nombre_evento (Text), Fecha_evento (Date), Nombre_Responsable (Text), Apellido_responsable (Text), Torre (Numeric), Apartamento (Numeric).
Salida:	Interfaz donde se muestre la información de la reserva del salón comunal.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Sistema] --&gt; UC1(Consulta del salón comunal)     Residente[Residente] --&gt; UC1     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(Ingresar al perfil módulo Salon Comunal)     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3(Consultar la tabla de las reservas para el salon) </pre>

Número:	R_12
Requisito:	Alquiler del salón comunal
Actor:	Administrador del conjunto y residentes
Descripción:	El administrador y el residente pueden reservar el salón comunal.
Proceso:	Los residentes y el administrador pueden proceder a reservar el uso del salón comunal deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar a los módulos de Salón comunal, en donde se diligencia un formulario de fecha y evento. El sistema le permitirá ingresar la información para realizar la reserva.
Entrada:	Número_evento (Numeric), Nombre_evento (Text), Fecha_evento (Date), Nombre_Responsable (Text), Apellido_responsable (Text), Torre (Numeric), Apartamento (Numeric).
Salida:	Interfaz donde se muestre el formulario con la información de la reserva del salón comunal.



Número:	R_13
Requisito:	Publicar informes
Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, puede publicar documentos de informes que sean requeridos para la difusión de información para la comunidad.
Proceso:	El usuario administrador del conjunto, debe haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de Informes donde puede seleccionar un documento, para luego poderlo adjuntar.
Entrada:	Dirección del archivo (text).
Salida:	Interfaz donde se muestren los archivos listados.
Diagrama Caso de Uso:	<pre> graph TD     AS[Administrador_Sistema] --&gt; UC1(Publicar informes)     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(Cargar el archivo de informe)     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3(Botón Publicar informe) </pre>

Número:	R_14
---------	------



Requisito:	Consultar y descargar informes
Actor:	Administrador del conjunto y residentes.
Descripción:	Los perfiles de administrador y residente del conjunto, puede consultar y descargar los informes publicados con los documentos de informes los cuales difunden la información para la comunidad.
Proceso:	Los usuarios administrador y residente del conjunto, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de Informes donde puede seleccionar un link de documento, para luego poder descargarlo.
Entrada:	Botón descargar informe. Dirección del archivo (text).
Salida:	Interfaz donde se muestren los archivos listados para consultarlos y descargarlos.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1([Consulta y descarga de los informes])     Residente[Residente] --&gt; UC1     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2([Ingresar al sistema módulo Informes])     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3([Consulta y descarga los informes que estan en el sistema]) </pre>

Número:	R_15
Requisito:	Consultar cartelera social
Actor:	Residentes y administrador del sistema
Descripción:	Los perfiles residente y administrador del sistema podrán acceder al módulo de Cartelera social que es alimentada por el administrador.
Proceso:	Los residentes y el administrador pueden proceder a la consulta información de la cartelera social, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al módulo de Cartelera social, en donde se realizará las respectivas consultas. El sistema le permitirá navegar mostrando la información alimentada por el administrador.
Entrada:	Nombre_Cartelera_social (Text)

	Contenido_Cartelera_social (Text)
Salida:	Interfaz donde se muestre la información de la Cartelera social.
Diagrama de Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1(Consultar información de la cartelera social)     Residente[Residente] --&gt; UC1     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(Ingresar al sistema módulo cartelera social)     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3(Consulta de la información de la cartelera social) </pre>

Número:	R_16
Requisito:	Editar información de la cartelera social
Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, puede editar la información de la cartelera social que haya creado para el conjunto cambiar nombre y cuerpo de la información de la cartelera social.
Proceso:	El usuario administrador del conjunto, debe haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo de cartelera social donde puede modificar la información de la cartelera social. Para ello se deben modificar algunos de los campos de la información.
Entrada:	Nombre_Cartelera_social (Text) Contenido_Cartelera_social (Text)
Salida:	Interfaz donde se muestre la información de la cartelera social.
Diagrama de Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1(Editar información de la cartelera social)     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(Ingresar datos)     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3(Boton actualizar)     UC3 -.-&gt; &lt;&lt;include&gt;&gt;  UC4(Publicar información actualizada) </pre>

Número:	R_17
Requisito:	Consultar Empresa de aseo.
Actor:	Administrador del conjunto y residente
Descripción :	El administrador y residente pueden consultar la información de la empresa de aseo.
Proceso:	La consulta de la Empresa de aseo en el sistema, se realiza por parte del administrador y residente del conjunto. Ingresa a su sesión de trabajo e ingresa al módulo de empresa de aseo, donde se ve el formulario con los datos de la empresa.
Entrada:	Nit Empresa (Numeric), Razón Social (Text), Dirección (Text), Número Telefónico1 (Numeric), Celular (Numeric), Estado de servicio(Boolean)
Salida:	Para el usuario administrador y residente del conjunto, se generara una Interfaz donde se muestra la información de la empresa de aseo registrada en el sistema.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1(Consultar empresa de aseo)     Residente[Residente] --&gt; UC1     UC1 -- "«include»" --&gt; UC2(Ingresar al perfil módulo empresa de aseo)     UC2 -- "«include»" --&gt; UC3(Consultar los datos de la empresa de aseo) </pre>

Número:	R_18
Requisito:	Actualizar datos Empresa de aseo
Actor:	Administrador del conjunto.
Descripción:	El usuario con perfil administrador de conjunto podrá actualizar algunos datos específicos de la empresa de aseo.
Proceso:	El usuario administrador del conjunto procederá a actualizar la información de la empresa de aseo por si hay algún cambio. Ingresa al modulo Empresa de aseo, en donde se realizará la respectiva modificación de los datos que han sido ingresados, según el criterio de la empresa. El sistema guardará las

	modificaciones realizadas, actualizando los datos modificados.
Entrada:	Nit Empresa (Numeric), Razón Social (Text), Dirección (Text), Número Telefónico1 (Numeric), Celular (Numeric), Estado de servicio(Boolean).
Salida:	Interfaz donde se muestre la información de la empresa de aseo donde se muestran los datos actualizados.
Diagrama Caso de Uso:	<pre> graph LR     Admin[Administrador_Conjunto] --&gt; UC1(Actualizar Datos Empresa Vigilancia)     UC1 -.-&gt; «include»  UC2(Modificar Datos de la Empresa Vigilancia)     UC2 -.-&gt; «include»  UC3(Guardar información Modificada) </pre>

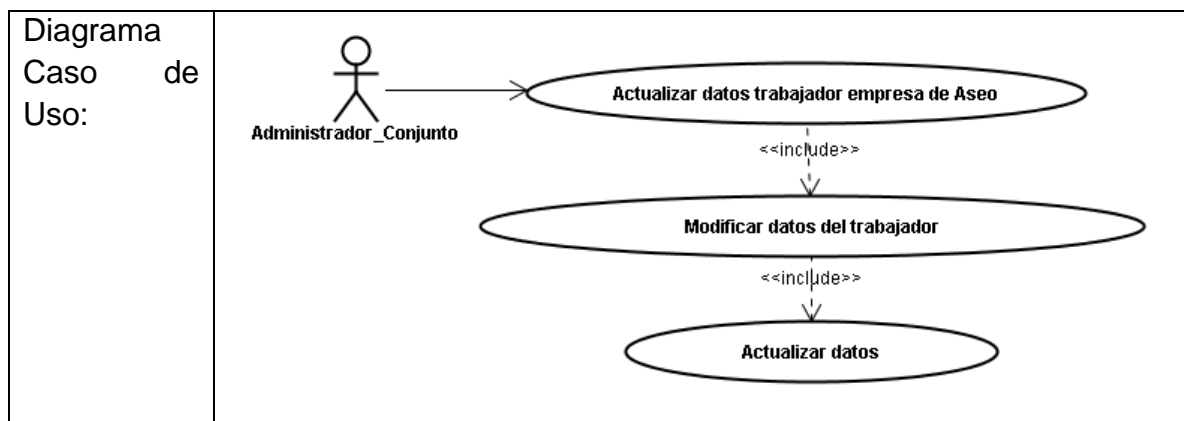
Número:	R_19
Requisito:	Consultar Empresa de vigilancia.
Actor:	Administrador del conjunto y residente
Descripción :	El administrador y residente pueden consultar la información de la empresa de vigilancia.
Proceso:	La consulta de la Empresa de vigilancia en el sistema, se realiza por parte del administrador y residente del conjunto. Ingresa a su sesión de trabajo e ingresa al módulo de vigilancia, donde se ve el formulario con los datos de la empresa.
Entrada:	Nit Empresa (Numeric), Razón Social (Text), Dirección (Text), Número Telefónico1 (Numeric), Celular (Numeric), Estado de servicio(Boolean)
Salida:	Para el usuario administrador y residente del conjunto, se generara una Interfaz donde se muestra la información de la empresa de vigilancia registrada en el sistema.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1(Consultar empresa de vigilancia)     Residente[Residente] --&gt; UC1     UC1 -.-&gt; «include»  UC2(Ingresar al perfil módulo empresa de vigilancia)     UC2 -.-&gt; «include»  UC3(Consultar los datos de la empresa de vigilancia) </pre>

Número:	R_20
Requisito:	Actualizar datos Empresa de vigilancia
Actor:	Administrador del conjunto
Descripción:	El usuario con perfil administrador de conjunto podrá actualizar algunos datos específicos de la empresa de vigilancia.
Proceso:	El usuario administrador del conjunto procederá a actualizar la información de la empresa de vigilancia por si hay algún cambio. Ingresa al modulo Empresa de Vigilancia, en donde se realizará la respectiva modificación de los datos que han sido ingresados, según el criterio de la empresa. El sistema guardará las modificaciones realizadas, actualizando los datos modificados.
Entrada:	Nit Empresa (Numeric), Razón Social (Text), Dirección (Text), Número Telefónico1 (Numeric), Celular (Numeric), Estado de servicio(Boolean).
Salida:	Interfaz donde se muestre la información de la empresa de vigilancia donde se muestran los datos actualizados.
Diagrama Caso de Uso:	<pre> graph LR     Actor[Administrador_Conjunto] --&gt; UC1(Actualizar Datos Empresa Vigilancia)     UC1 -.-&gt; ≤ &lt;&lt;include&gt;&gt;  UC2(Modificar Datos de la Empresa Vigilancia)     UC2 -.-&gt; ≤ &lt;&lt;include&gt;&gt;  UC3(Guardar información Modificada) </pre>

Número:	R_21
Requisito:	Consultar información de trabajadores empresa de aseo
Actor:	Administrador del conjunto y residentes
Descripción :	El perfil de administrador y residente del conjunto puede consultar los datos del personal de aseo.
Proceso:	El usuario Administrador y residente del conjunto puede consultar la información del trabajador de la empresa de aseo, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo Empresa de aseo. En donde se realizará la respectiva consulta de los datos de los trabajadores de la empresa.
Entrada:	Numero de documento (Numeric), Nombres (Text), Apellidos (Text), Dirección (Text), Número Telefónico1 (Numeric), Número Telefónico2 (Numeric), Celular (Numeric) y Estado de servicio(Boolean)

Salida:	Interfaz donde se muestra la información del trabajador registrado por el administrador del conjunto.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1([Consulta del trabajador de la empresa de aseo])     Residente[Residente] --&gt; UC1     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2([Ingresa al perfil módulo empresa de aseo])     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3([Consultar los datos de los trabajadores de la empresa de aseo]) </pre>

Número:	R_22
Requisito:	Actualizar datos de los Trabajadores de la empresa de aseo.
Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, puede actualizar los datos del personal de aseo, cada vez que se cambie un trabajador.
Proceso:	Los usuarios Administrativos para proceder a actualizar la información de trabajadores de la empresa de aseo, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo Empresa de aseo, en donde se realizará la respectiva modificación de los datos que han sido ingresados, según el criterio del usuario. El sistema guardará las modificaciones realizadas, actualizando los datos modificados.
Entrada:	Numero de documento (Numeric), Nombres (Text), Apellidos (Text), Dirección (Text), Número Telefónico1 (Numeric), Número Telefónico2 (Numeric), Celular (Numeric) y Estado de servicio(Boolean)
Salida:	Interfaz donde se muestre la información de los trabajadores de la empresa de aseo, incluyendo las modificaciones realizadas.



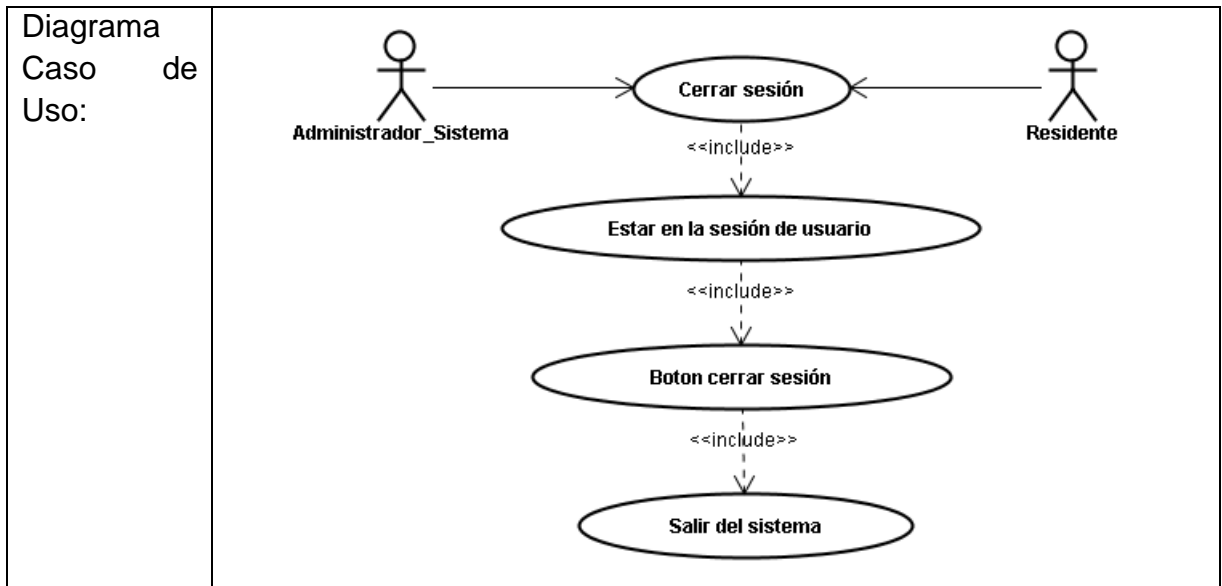
Número:	R_23
Requisito:	Consultar información de trabajadores empresa de vigilancia
Actor:	Administrador del conjunto y residentes
Descripción :	El perfil de administrador y residente del conjunto puede consultar los datos del personal de vigilancia.
Proceso:	El usuario Administrador y residente del conjunto puede consultar la información del trabajador de la empresa de vigilancia, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo Empresa de vigilancia. En donde se realizará la respectiva consulta de los datos de los trabajadores de la empresa.
Entrada:	Numero de documento (Numeric), Nombres (Text), Apellidos (Text), Dirección (Text), Número Telefónico1 (Numeric), Número Telefónico2 (Numeric), Celular (Numeric) y Estado de servicio(Boolean)
Salida:	Interfaz donde se muestra la información del trabajador registrado por el administrador del conjunto.
Diagrama Caso de Uso:	<pre> graph TD     Admin[Administrador_Conjunto] --&gt; UC1(Consulta del trabajador de la empresa de aseo)     Residente[Residente] --&gt; UC1     UC1 -.-&gt; &lt;&lt;include&gt;&gt;  UC2(Ingresa al perfil módulo empresa de aseo)     UC2 -.-&gt; &lt;&lt;include&gt;&gt;  UC3(Consultar los datos de los trabajadores de la empresa de aseo) </pre>

Número:	R_24
---------	------

Requisito:	Actualizar datos de los Trabajadores de la empresa de vigilancia.
Actor:	Administrador del conjunto.
Descripción:	El perfil de administrador del conjunto, puede actualizar los datos del personal de aseo, cada vez que se cambie un trabajador.
Proceso:	Los usuarios Administrativos y residentes para proceder a actualizar su información, deben haber ingresado al sistema en su respectiva sesión de trabajo, e ingresar al modulo Empresa de aseo, en donde se realizará la respectiva modificación de los datos que han sido ingresados, según el criterio del usuario. El sistema guardará las modificaciones realizadas, actualizando los datos modificados.
Entrada:	Numero de documento (Numeric), Nombres (Text), Apellidos (Text), Dirección (Text), Número Telefónico1 (Numeric), Número Telefónico2 (Numeric), Celular (Numeric) y Estado de servicio(Boolean)
Salida:	Interfaz donde se muestre la información de los trabajadores de la empresa de vigilancia actualizados
Diagrama Caso de Uso:	<pre> graph TD     Actor[Administrador_Conjunto] --&gt; UC1(Actualizar datos trabajador empresa de Vigilancia)     UC1 -- "&lt;&lt;include&gt;&gt;" --&gt; UC2(Modificar datos del trabajador)     UC2 -- "&lt;&lt;include&gt;&gt;" --&gt; UC3(Actualizar datos) </pre>

Número:	R_25
Requisito:	Cerrar sesión
Actor:	El administrador del Conjunto y los residentes del conjunto.
Descripción:	El software administrador de conjuntos cerrados brindará acceso a éste ingresando los datos y así mismo el cerrar sesión del aplicativo una vez termine sus operaciones o consulta, por seguridad.
Proceso:	Para salir del sistema, es necesario que el usuario se encuentre registrado en el mismo e ingrese al sistema y realice su operación normal, luego valla al botón cerrar sesión y salga del aplicativo.
Entrada:	Número de Identificación (Numeric)
Salida:	Interfaz con la opción de salir del sistema.





#### 4.1.3. REQUERIMIENTOS NO FUNCIONALES.

- La información de manejo de propietarios, administrativos y personal de las empresas de servicio al conjunto cerrado debe ser privada y confidencial al público en general. Solo debe ser propiedad del usuario y la administración. No se revelará información personal de los clientes.
- Se requiere disponibilidad de un alojamiento web para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web de la aplicación del proyecto.
- Los correos que va a manejar el administrador y los residentes son de GMAIL.

#### 4.1.4. CRONOGRAMA.

TABLA2: CRONOGRAMA.

1		<b>SIACC</b>	<b>455 days?</b>	<b>3/01/11 08:00 AM</b>	<b>28/09/12 05:00 PM</b>	
2		<b>INICIACIÓN</b>	<b>60 days?</b>	<b>3/01/11 08:00 AM</b>	<b>25/03/11 05:00 PM</b>	
3		Definición de requerimientos	15 days	3/01/11 08:00 AM	21/01/11 05:00 PM	
4		Lista de requerimientos funcionales y no funcionales	15 days?	24/01/11 08:00 AM	11/02/11 05:00 PM	
5		Definición de propuesta de alcance del proyecto	15 days	14/02/11 08:00 AM	4/03/11 05:00 PM	
6		Levantamiento de información-Herramientas	15 days?	7/03/11 08:00 AM	25/03/11 05:00 PM	5
7		<b>FASE DE ANALISIS</b>	<b>30 days?</b>	<b>18/04/11 08:00 AM</b>	<b>27/05/11 05:00 PM</b>	
8		Estudio preliminar- Herramientas para el desarrollo	15 days?	18/04/11 08:00 AM	6/05/11 05:00 PM	5
9		Definición de Requerimientos Funcionales-No Funcionales Sist...	15 days?	9/05/11 08:00 AM	27/05/11 05:00 PM	
10		<b>FASE DE DISEÑO</b>	<b>115 days?</b>	<b>30/05/11 08:00 AM</b>	<b>4/11/11 05:00 PM</b>	
11		Prototipos Funcionales-Investigación y pruebas	15 days?	30/05/11 08:00 AM	17/06/11 05:00 PM	
12		Definición de arquitectura Física	15 days?	4/07/11 08:00 AM	22/07/11 05:00 PM	
13		Definición de arquitectura lógica	15 days?	25/07/11 08:00 AM	12/08/11 05:00 PM	
14		Definición de funcionamiento del sistema	15 days?	15/08/11 08:00 AM	2/09/11 05:00 PM	
15		Arquitectura de la solución	15 days?	5/09/11 08:00 AM	23/09/11 05:00 PM	
16		Diagrama de contexto	15 days?	26/09/11 08:00 AM	14/10/11 05:00 PM	
17		Presentación de prototipo funcional base para cliente	15 days?	17/10/11 08:00 AM	4/11/11 05:00 PM	
18		<b>FASE DE DESARROLLO</b>	<b>180 days</b>	<b>14/11/11 08:00 AM</b>	<b>20/07/12 05:00 PM</b>	
19		<b>Modulo Residente</b>	<b>40 days</b>	<b>14/11/11 08:00 AM</b>	<b>6/01/12 05:00 PM</b>	
20		Interfaz	20 days	14/11/11 08:00 AM	9/12/11 05:00 PM	
21		Funcionalidad	20 days	12/12/11 08:00 AM	6/01/12 05:00 PM	
22		<b>Modulo Administrador</b>	<b>40 days</b>	<b>9/01/12 08:00 AM</b>	<b>2/03/12 05:00 PM</b>	
23		Interfaz	20 days	9/01/12 08:00 AM	3/02/12 05:00 PM	
24		Funcionalidad	20 days	6/02/12 08:00 AM	2/03/12 05:00 PM	
25		<b>Modulo Empresas</b>	<b>40 days</b>	<b>5/03/12 08:00 AM</b>	<b>27/04/12 05:00 PM</b>	
26		Interfaz	20 days	5/03/12 08:00 AM	30/03/12 05:00 PM	
27		Funcionalidad	20 days	2/04/12 08:00 AM	27/04/12 05:00 PM	
28		<b>Pruebas</b>	<b>60 days</b>	<b>30/04/12 08:00 AM</b>	<b>20/07/12 05:00 PM</b>	
29		Pruebas de captura de datos	20 days	30/04/12 08:00 AM	25/05/12 05:00 PM	
30		Pruebas de sincronización	20 days	28/05/12 08:00 AM	22/06/12 05:00 PM	
31		Pruebas de referenciación de información en reportes	20 days	25/06/12 08:00 AM	20/07/12 05:00 PM	
32		<b>FASE DE IMPLEMENTACION-CAPACITACION</b>	<b>50 days?</b>	<b>23/07/12 08:00 AM</b>	<b>28/09/12 05:00 PM</b>	
33		Creación de documentación del producto terminado	27 days	23/07/12 08:00 AM	28/08/12 05:00 PM	
34		Modificaciones segun sugerencias	7 days?	14/09/12 08:00 AM	24/09/12 05:00 PM	
35		Capacitación del producto terminado	4 days	25/09/12 08:00 AM	28/09/12 05:00 PM	
36		Presentación del producto entregado	1 day?	28/09/12 08:00 AM	28/09/12 05:00 PM	
SIACC - página 1						

Fuente: Autores

## 4.2. CAPITULO 2 DISEÑO:

### 4.2.1. Diseño de arquitectura

IMAGEN 12. ARQUITECTURA DEL SISTEMA

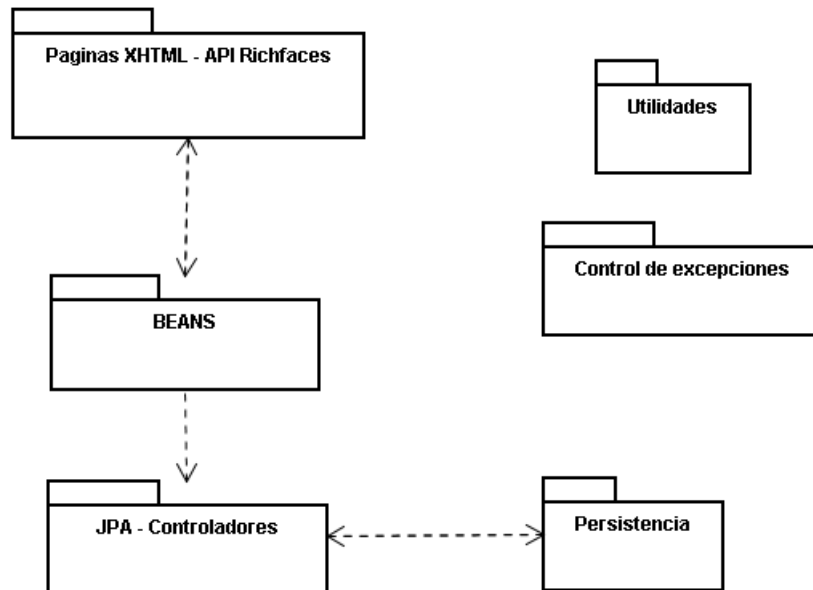


Fuente: Autores

**Imagen 12. Arquitectura del sistema:** En la imagen se visualiza la arquitectura que se usó en el desarrollo del aplicativo SIACC.

#### 4.2.2. Diseño de paquetes

IMAGEN 13. DISEÑO DE PAQUETES

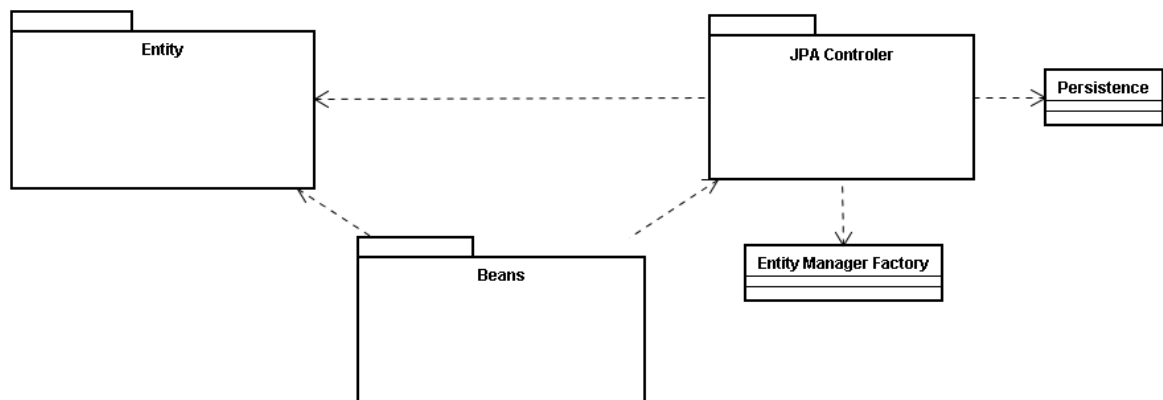


Fuente: Autores

**Imagen 13. Diseño de paquetes:** En la imagen se visualiza el diseño de paquetes con el cual se realizó el aplicativo SIACC.

#### 4.2.3. Diseño de clases

IMAGEN 14. DIAGRAMA DE CLASES GENERAL



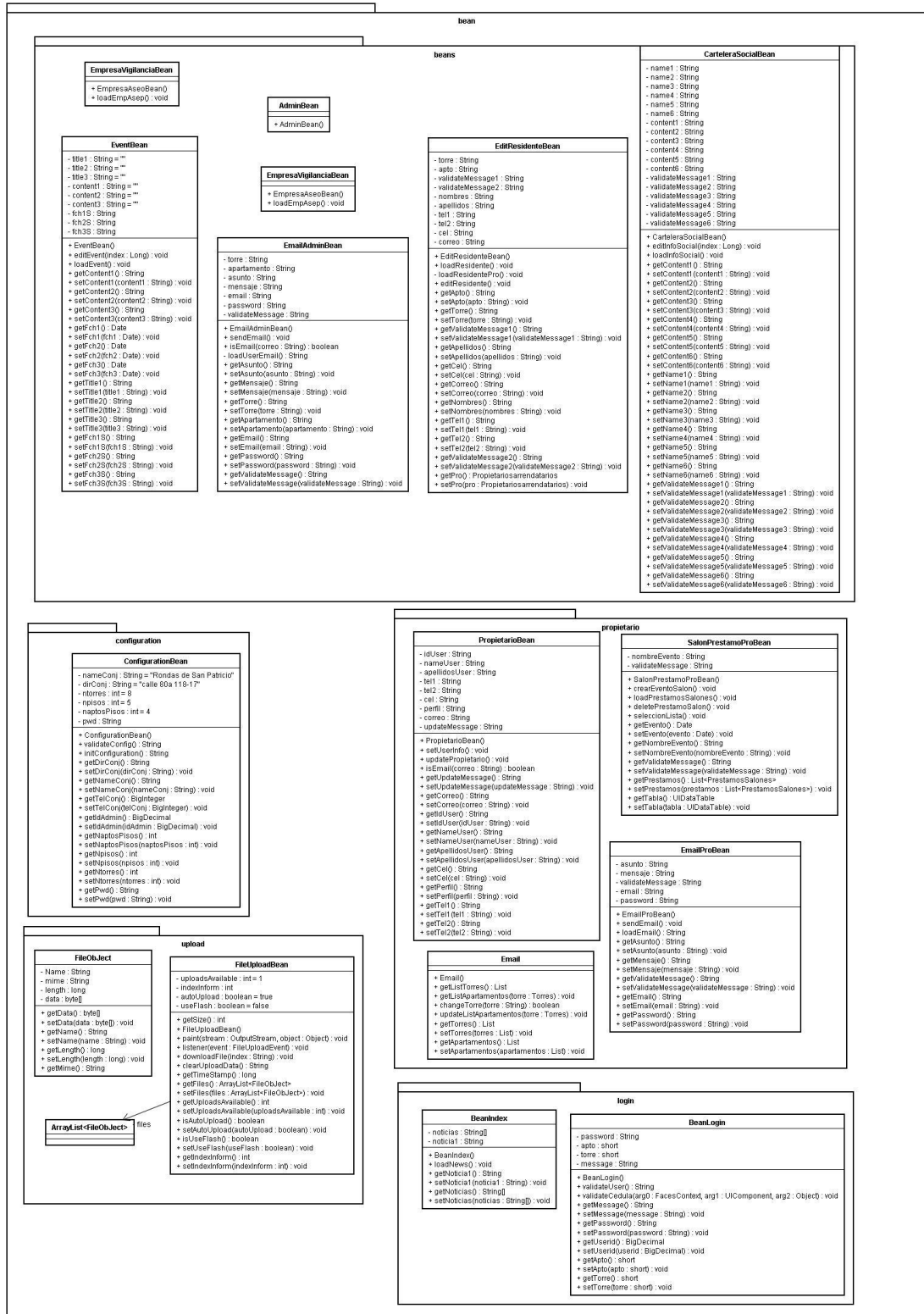
Fuente: Autores

IMAGEN 15. DIAGRAMA DE CLASES ENTITY



~ 77 ~

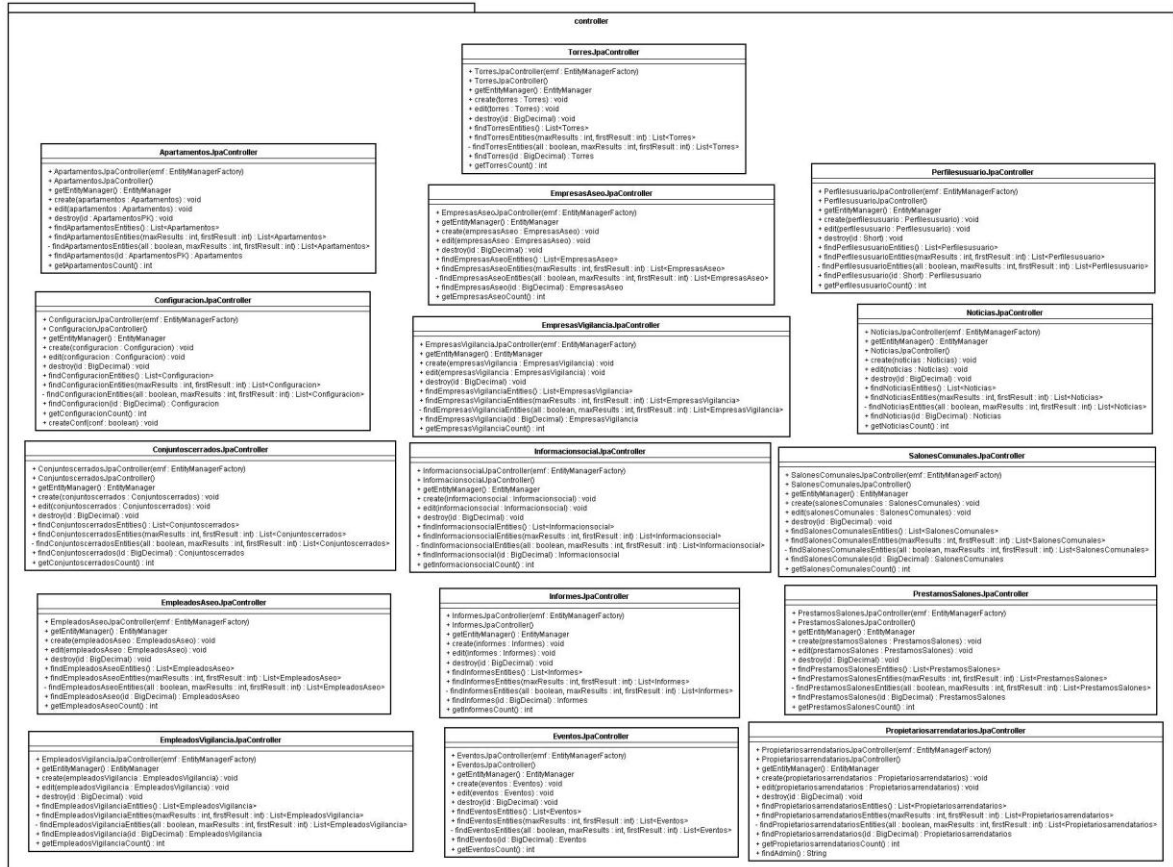
## IMAGEN 16. DIAGRAMA DE CLASES BEANS



Fuente: Autores

**Imagen 16. Diagrama de clases Entity:** En la imagen se visualiza el diagrama de clases enfocado únicamente en el Entity.

IMAGEN 17. DIAGRAMA DE CLASES CONTROLER



Fuente: Autores

**Imagen 17. Diagrama de clases Controler:** En la imagen se visualiza el diagrama de clases enfocado únicamente en el Controler.

#### 4.2.4. Diseño de pruebas

Se realiza un diseño con 74 casos de prueba los cuales pasan satisfactoriamente.  
ANEXO 6. DISEÑO DE LAS PRUEBAS.

Se realizan las pruebas de despliegue y de sistema con los siguientes formatos (anexo 8):

**PRUEBAS DE INTEGRACION.**

Escriba nombre del proyecto					
<b>CÓDIGO CASO</b>	Nombre	<b>VERSIÓN</b>	2.0	<b>FECHA</b>	JUN/2012

PRUEBAS DE INTEGRACION							
<b>MÓDULO</b>	Escriba nombre modulo	INTERFA CE o CLASE	Escriba el nombre de la clase o interface	<b>VERSIÓN</b>	1. 0	<b>ING DESARROLLO (programador)</b>	Miguel Ángel Colorado.
<b>DESCRIPCIÓN</b>	Escriba el contexto de la prueba						
<b>OBJETIVOS</b>	Escriba objetivo de las pruebas						
<b>REQUERIMIENTOS</b>	Escriba el nombre de los requerimientos técnicos para realizar la prueba. Ejemplo <ul style="list-style-type: none"> <li>Jdk 1.6</li> <li>Jar de dom4j</li> <li>Postgresql</li> </ul>						
<b>PRE CONDICIONES</b>	Escriba las condiciones para la realización de la prueba. Ejemplo : Motor de base de datos activo con la conexión establecida						
DETALLES DE LA PRUEBA							
<b>EJECUCIÓN Nº</b>	1	<b>FECHA DE LA EJECUCION</b>	21/09/2012	<b>ING PRUEBAS</b>	Yina Paola Chaves Silva.		
<b>OBSERVACIONES</b>							
<b>Nº</b>	<b>MÉTODO</b>	<b>PARÁMETROS</b>	<b>RESULTADO ESPERADO</b>	<b>RESULTADO OBTENIDO</b>	<b>CALIFICACION</b>	<b>OBSERVACIONES</b>	
					A <input type="checkbox"/> R <input type="checkbox"/>		
					A <input type="checkbox"/> R <input type="checkbox"/>		



**PRUEBAS DE SISTEMA.**

<b>Escriba nombre del proyecto</b>					
<b>CÓDIGO CASO</b>	Nombre	<b>VERSIÓN</b>	2.0	<b>FECHA</b>	JUN/2012

PRUEBAS DE SISTEMA						
Caso de Uso	Escriba nombre caso de uso		VERSIÓN	1.0	ING DESARROLLO (programador)	Miguel Ángel Colorado.
DESCRIPCIÓN	Escriba el contexto de la prueba					
OBJETIVOS	Escriba objetivo de las pruebas					
REQUERIMIENTOS	Escriba el nombre de los requerimientos técnicos para realizar la prueba. Ejemplo <ul style="list-style-type: none"> <li>Jdk 1.6</li> <li>Jar de dom4j</li> <li>Postgresql</li> </ul>					
PRE CONDICIONES	Escriba las condiciones para la realización de la prueba. Ejemplo : Motor de base de datos activo con la conexión establecida					
DETALLES DE LA PRUEBA						
EJECUCIÓN Nº	1	FECHA DE LA EJECUCION	21/09/2012	ING PRUEBAS	Yina Paola Chaves Silva.	
OBSERVACIONES						
Nº	FUNCIONALIDAD ESPECÍFICA	PARÁMETROS	RESULTADO ESPERADO	RESULTADO OBTENIDO	CALIFICACION	OBSERVACIONES
					A <input type="checkbox"/> R <input type="checkbox"/>	
					A <input type="checkbox"/> R <input type="checkbox"/>	

### **4.3. CAPITULO 3. IMPLEMENTACIÓN.**

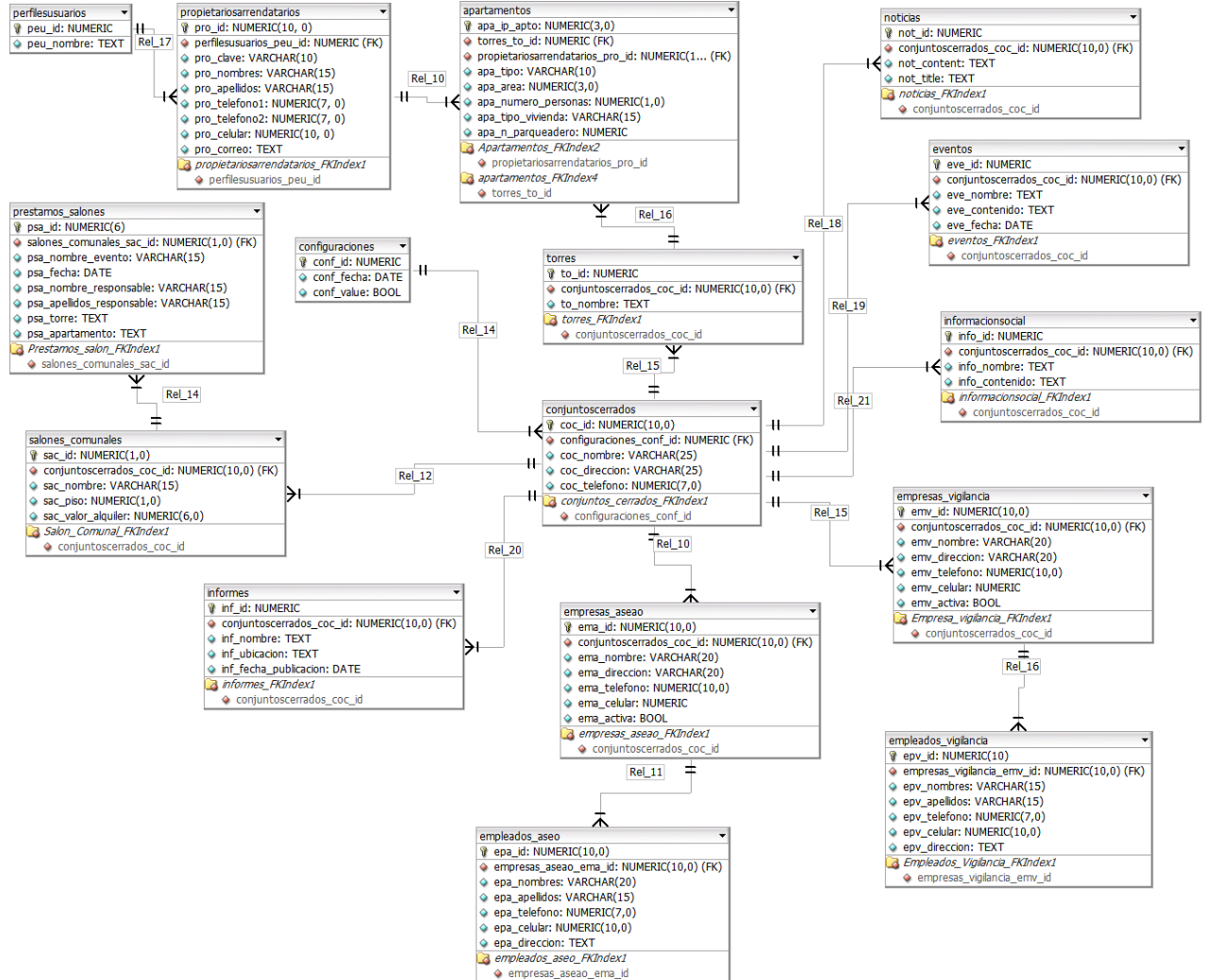
El presente capítulo, se explica y describe los puntos del proyecto que se tuvieron en cuenta para la implementación desde la base de soporte de datos y de qué manera el usuario final realizaría su interacción con el software.

Para la implementación de software determinamos las estructuras de datos y las representamos en el siguiente punto a través de un Modelo entidad relación, el cual representamos lógicamente las entidades modeladas para el soporte en datos para el proyecto SIACC.

En este proceso necesitábamos dar a conocer el prototipo para la implementación al usuario final a través de unos wireframes predefinidos, los cuales son la pauta inicial para el desarrollo de la interfaz gráfica y la deducción del flujo funcional de la aplicación software.

#### 4.3.1. DIAGRAMA DE MER.

IMAGEN 18. MODELO ENTIDAD RELACIÓN



Fuente: Autores

**Imagen 18. Modelo entidad relación:** En la imagen se visualiza el Modelo entidad relación del proyecto SIACC, con todas las entidades de la BD.

#### 4.3.2. Estándares y prototipos de las pantallas del sistema.

##### CARGUE INICIAL

IMAGEN 19. PANTALLA DE CARGUE

Diagrama de la interfaz de usuario para la pantalla de carga inicial. El diseño incluye un botón de "CARGUE" en la parte superior. Debajo, hay tres pares de campos de entrada etiquetados como "Número de Torres:", "Número de pisos:" y "Número aptos por piso:", cada uno con un "Textbox" adyacente. Más abajo, se encuentran los campos de "Identificación Administrador:" y "Contraseña del Adminsitrador:" (con una ortografía incorrecta en el prototipo), también con "Textbox" adyacentes. En la parte inferior del formulario, hay un botón etiquetado como "Configuración Conjunto".

Fuente: Autores

**Imagen 19. Pantalla de cargue:** En la imagen se visualiza el Modelo de la pantalla inicial de cargue del aplicativo Web SIACC.

##### LOGIN

IMAGEN 20. PANTALLA DEL LOGIN

Diagrama de la interfaz de usuario para la pantalla de inicio de sesión. El diseño comienza con un botón de "INICIAR SESIÓN" en la parte superior. A continuación, hay dos pares de campos de entrada etiquetados como "IDENTIFICACIÓN:" y "CLAVE:", cada uno con un campo de texto adyacente. En la parte inferior del formulario, hay un botón etiquetado como "INICIAR SESIÓN".

Fuente: Autores

**Imagen 20. Pantalla de login:** En la imagen se visualiza el Modelo de la pantalla para realizar el logeo del residente o administrador en el aplicativo Web SIACC.

## ADMINISTRADOR – PERFIL

IMAGEN 21. PANTALLA DEL PERFIL DEL ADMINISTRADOR

PERFIL - ASMINISTRADOR

DATOS DEL ADMINISTRADOR

Nombres: Textbox

Apellidos: Textbox

Telefono fijo 1: Textbox

Telefono fijo 2: Textbox

Telefono Celular: Textbox

Tipo de Residente: Textbox

Guardar Cancelar

Fuente: Autores

**Imagen 21. Pantalla del perfil del administrador:** En la imagen se visualiza el Modelo de la pantalla y opciones de perfil que tiene el administrador en el aplicativo Web SIACC.

## ADMINISTRADOR – NOTICIAS

IMAGEN 22. PANTALLA DE LAS NOTICIAS EN ADMINISTRADOR

NOTICIAS ADMINISTRADOR

NOTICIA 1

Nombre de la noticia: Textbox

Cuerpo de la noticia: Textbox

Editar Noticia Cancelar Edición

Fuente: Autores

**Imagen 22. Pantalla de las noticias en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de editar noticias desde perfil administrador en el aplicativo Web SIACC.

## ADMINISTRADOR – EVENTOS

IMAGEN 23. PANTALLA DE EVENTOS EN ADMINISTRADOR

El diagrama muestra la estructura de la interfaz de usuario para la pantalla de eventos en administrador. Comienza con un botón azul rectangular con el texto "EVENTOS ADMINISTRADOR". Debajo de este hay un botón gris rectangular con el texto "EVENTO 1". A continuación, se encuentra un grupo de entrada etiquetado "Nombre del Evento:" que incluye un campo de texto rectangular. Después, hay otro grupo de entrada etiquetado "Cuerpo del Evento:" que incluye un campo de texto rectangular más grande. Al final, hay dos botones rectangulares: "Editar Evento" a la izquierda y "Cancelar Edición" a la derecha.

Fuente: Autores

**Imagen 23. Pantalla de eventos en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de editar eventos desde perfil administrador en el aplicativo Web SIACC.

## ADMINISTRADOR – CORREO

IMAGEN 24. PANTALLA DE CORREO EN ADMINISTRADOR

NOTICIAS - ADMINSTRADOR

Seleccionar propietario a enviar correo :

Torre : Textbox

Apartamento : Textbox

Cuerpo del mensaje : Textbox

Enviar Cancelar Envío

Fuente: Autores

**Imagen 24. Pantalla de correo en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de correo desde el perfil administrador en el aplicativo Web SIACC.

**ADMINISTRADOR – SALONCOMUNAL**

IMAGEN 25. PANTALLA DE SALÓN COMUNAL EN ADMINISTRADOR



Fuente: Autores

**Imagen 25. Pantalla de salón comunal en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de salón comunal desde el perfil administrador en el aplicativo Web SIACC.



## ADMINISTRADOR – INFORMES

IMAGEN 26. PANTALLA INFORMES EN ADMINISTRADOR

Diagrama de la interfaz de usuario para la pantalla de informes en el perfil de administrador. Se muestran dos secciones principales:

- INFORMES - ADMINISTRADOR** (encabezado azul):
  - INFORME GENERAL** (botón gris):
    - Botón **Add** (pequeño botón gris dentro de un recuadro azul).
  - INFORME DETALLADO** (botón gris):
    - Botón **Add** (pequeño botón gris dentro de un recuadro azul).

Fuente: Autores

**Imagen 26. Pantalla informes en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de cargar los informes del conjunto desde el perfil administrador en el aplicativo Web SIACC.

## ADMINISTRADOR –CARTELERA SOCIAL

IMAGEN 27. PANTALLA CARTELERA SOCIAL EN ADMINISTRADOR

Diagrama de la interfaz de usuario para la pantalla de cartelera social en el perfil de administrador. Se muestran los siguientes elementos:

- CARTELERA SOCIAL ADMINISTRADOR** (encabezado azul).
- INFORMACIÓN 1** (encabezado gris).
- Formulario de entrada:
  - Nombre de la Información:** (etiqueta) y **Textbox** (campo de entrada).
  - Cuerpo de la Información:** (etiqueta) y **Textbox** (campo de entrada de mayor tamaño).
- Botones de acción:
  - Editar Información** (botón gris).
  - Cancelar Edición** (botón gris).

Fuente: Autores

**Imagen 27. Pantalla cartelera social en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de edición de la información de cartelera social desde el perfil administrador en el aplicativo Web SIACC.

## ADMINISTRADOR – MODIFICAR PROPIETARIO

IMAGEN 28. PANTALLA MODIFICAR PROPIETARIO EN ADMINISTRADOR

MODIFICAR PROPIETARIO - ADMINISTRADOR

Datos del propietario a modificar :

Número de torre :  Textbox

Número de apartamento:  Textbox

Cargar Información

Datos del propietario a modificar :

Torre :  #

Apartamento :  #

Área Apartamento :  #

Número de parqueadero :  #

Nombres :  Textbox

Apellidos :  Textbox

Telefono fijo 1 :  Textbox

Telefono fijo 2 :  Textbox

Telefono celular :  Textbox

Tipo de residente :  Textbox

Actualizar Cancelar

Fuente: Autores

**Imagen 28. Pantalla modificar propietario en administrador:** En la imagen se visualiza el Modelo de la pantalla con las opciones de edición de la información del propietario desde el perfil administrador en el aplicativo Web SIACC.

## ADMINISTRADOR –CERRAR SESIÓN

IMAGEN 29. PANTALLA CERRAR SESIÓN EN ADMINISTRADOR

MENU PRINCIPAL ADMINISTRADOR

Cerrar Sesión

PERFIL NOTICIAS EVENTOS CORREO SALÓN COMUNAL INFORMES CARTELERA SOCIAL MODIFICAR PROPIETARIO

Fuente: Autores

**Imagen 29. Pantalla cerrar sesión en administrador:** En la imagen se visualiza el Modelo de la pantalla con la opción de cerrar sesión desde el perfil administrador en el aplicativo Web SIACC.

## DATOS – PROPIETARIO

IMAGEN 30. PANTALLA DATOS DEL PERFIL EN PROPIETARIO

DATOS - PROPIETARIO

PROPIETARIO

Cedula del propietario: Textbox

Nombre del propietario: Textbox

Torre : Textbox

Apartamento : Textbox

Cerrar Sesión.

Fuente: Autores

**Imagen 30. Pantalla datos del perfil en propietario:** En la imagen se visualiza el Modelo de la pantalla con la información del residente desde el perfil propietario en el aplicativo Web SIACC.

## NOTICIAS – PROPIETARIO

IMAGEN 31. PANTALLA DE NOTICIAS EN PROPIETARIO

NOTICIAS - PROPIETARIO

NOTICIA 1

Nombre de la noticia

Textbox

Cuerpo de la noticia

Textbox

Fuente: Autores

**Imagen 31. Pantalla de noticias en propietario:** En la imagen se visualiza el Modelo de la pantalla con la información de noticias del conjunto desde el perfil propietario en el aplicativo Web SIACC.

## EVENTOS – PROPIETARIO

IMAGEN 32. PANTALLA DE EVENTOS EN PROPIETARIO

Diagrama de la interfaz de usuario para la pantalla de eventos en propietario. El formulario comienza con un encabezado azul que dice "EVENTOS - PROPIETARIO". Debajo hay un botón gris que dice "EVENTO 1". El formulario principal tiene dos secciones: la primera con el label "Nombre del evento :" y un campo de texto "Textbox"; la segunda con el label "Cuerpo del evento :" y un campo de texto "Textbox" más grande.

Fuente: Autores

**Imagen 32. Pantalla de eventos en propietario:** En la imagen se visualiza el Modelo de la pantalla con la información de eventos del conjunto desde el perfil propietario en el aplicativo Web SIACC.

## CORREO – PROPIETARIO

IMAGEN 33. PANTALLA DE CORREO EN PROPIETARIO

Diagrama de la interfaz de usuario para la pantalla de correo en propietario. El formulario comienza con un encabezado azul que dice "CORREO - PROPIETARIO". Debajo hay un botón azul que dice "EDITAR CORREO". El formulario principal tiene tres campos de texto: "Dirección de correo :", "Asunto del correo :", y "Mensaje :". A la derecha de los primeros dos campos hay un campo de texto "Textbox". Debajo de los campos de texto hay un campo de texto "Textbox" más grande. En la parte inferior del formulario hay un botón gris que dice "Enviar mensaje".

Fuente: Autores

**Imagen 33. Pantalla de correos en propietario:** En la imagen se visualiza el Modelo de la pantalla con la opción de correo desde el perfil propietario en el aplicativo Web SIACC.

## SALÓN COMUNAL – PROPIETARIO

IMAGEN 34. PANTALLA DE SALÓN COMUNAL EN PROPIETARIO

Seleccionar fecha del alquiler:

dom	lun	mar	mié	jue	vie	sáb	dom
	27	28	29	30	1	2	3
4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27
28	29	30	31	1	2	3	4
5	6	7	8	9	10	11	12

Today

FECHA	NÚMERO DE TORRE	NÚMERO DE APARTAMENTO	NOMBRE DEL PROPIETARIO
Sample 1	Sample 1	Sample 1	Sample 1
Sample 2	Sample 2	Sample 2	Sample 2
Sample 3	Sample 3	Sample 3	Sample 3

Fuente: Autores

**Imagen 34. Pantalla de correos en propietario:** En la imagen se visualiza el Modelo de la pantalla con la opción de salón comunal, donde se ven las fechas ya seleccionadas y da la opción para poder apartar el salón, desde el perfil propietario en el aplicativo Web SIACC.

## INFORMES – PROPIETARIO

IMAGEN 35. PANTALLA DE INFORMES EN PROPIETARIO

Informe general

Informe general, clic aqui: [Informe general](#)

Informe detallado

Informe detallado, clic aqui: [Informe detallado](#)

Fuente: Autores

**Imagen 35. Pantalla de informes en propietario:** En la imagen se visualiza el Modelo de la pantalla con la opción de informes, donde se pueden descargar los informes contables del conjunto desde el perfil propietario en el aplicativo Web SIACC.

## CARTELERA SOCIAL – PROPIETARIO

IMAGEN 36. PANTALLA DE CARTELERA SOCIAL EN PROPIETARIO

CARTELERA SOCIAL - PROPIETARIO	
<div>Información 1</div> <div>Textbox</div>	<div>Información 2</div> <div>Textbox</div>
<div>Información 3</div> <div>Textbox</div>	<div>Información 4</div> <div>Textbox</div>
<div>Información 5</div> <div>Textbox</div>	<div>Información 6</div> <div>Textbox</div>

Fuente: Autores

**Imagen 36. Pantalla de cartelera social en propietario:** En la imagen se visualiza el Modelo de la pantalla con la opción de cartelera social, donde se puede ver la información social del conjunto desde el perfil propietario en el aplicativo Web SIACC.

## PERFIL – PROPIETARIO

IMAGEN 37. PANTALLA DE PERFIL EN PROPIETARIO

PERFIL DEL PROPIETARIO	
Torre :	Textbox
Apartamento :	Textbox
Área de apartamento :	Textbox
Número de parqueadero :	Textbox
Nombres :	Textbox
Apellidos :	Textbox
Telefono fijo 1 :	Textbox
Telefono fijo 2 :	Textbox
Telefono celular :	Textbox
Tipo de residente :	Textbox
<div>Actualizar Cancelar</div>	

Fuente: Autores

**Imagen 37. Pantalla de perfil en propietario:** En la imagen se visualiza el Modelo de la pantalla con la información del residente y la opción para modificar algún dato que se haya cambiado desde el perfil propietario en el aplicativo Web SIACC.

#### 4.3.3. RESULTADO DE PRUEBAS

Se realiza la prueba de despliegue y su resultado fue exitoso, teniendo en cuenta los prerequisites necesarios para el sistema, los cuales están en el marco tecnológico y son los que se muestran a continuación.

Se realiza un diseño con setenta y cuatro (74) casos de prueba para validar la parte funcional del aplicativo, los cuales se ejecutaron y se certifica, teniendo en cuenta en la base de datos 160 registros.

Se realizaron los casos de prueba para validación de casos de uso y pruebas para integración del sistema (Anexo 8).

Las características técnicas que se tuvieron en cuenta para la realización de las pruebas del software fueron las siguientes:

- Características del ambiente de pruebas a nivel de red:  
Velocidad de 4 Mb
- Características del ambiente de pruebas a nivel de hardware:  
Equipo portátil Hewlett Packard con Procesador Intel core i3 de 2.3 GHz, memoria RAM 4 GB, sistema operativo Windows 7 home Premium de software 32 bits y hardware 64 bits, disco duro de 360 Gb, Postgres 8.4, Apache Tomcat 7 y JDK 7.

#### **4.4. CAPITULO 4. IMPLANTACIÓN**

##### **4.4.1. INSTALACIÓN.**

Se realiza la instalación del software en un equipo Hewlett Packard con Procesador AMD Athlon™ II Dual – Core M320 de 2.10 GHz, Memoria RAM 3Gb, Sistema Operativo 32 bits, Disco Duro 360 Gb. Se tiene instalado previamente el motor de base de datos Postgres 8.4, el servidor de aplicaciones Web Apache tomcat de la versión 6.18 en adelante y JAVA instalado con el JDK 7.

Se realiza el ANEXO 5, Manual de instalación.



## **5. RESULTADOS Y DISCUSIÓN.**

Se obtiene un sistema el cual facilita la comunicación entre la administración y los residentes de un conjunto cerrado, un sistema novedoso en cuanto a que no se conoce de un software que ayude a la parte social de una población que se hace cada vez más grande en Bogotá, como lo es la construcción de conjuntos residenciales que abarca todo tipo de estratos y ubicaciones.

Por tal razón pensamos que este sistema además de hacerse con herramientas tecnológicas se hace con un fin social y es el de facilitar esta comunicación, teniendo en cuenta la problemática de los residentes la cual es el no permanecer en el conjunto en los horarios de atención de la administración, debido a sus horarios laborales.

## **6. Conclusiones.**

Teniendo en cuenta que una factor importante para un residente es la participación en el manejo de recursos y toma de decisiones del conjunto residencial donde vive, se implementa un sistema el cual ayude a hacer una brecha más pequeña entre la comunicación directa mas no personal del administrador y los residentes.

Además con este proyecto se puede ver que existe otra forma usando las Tics de tener informado al residente de todos los eventos, noticias o de más actividades que suceden a diario en el conjunto y de los cuales está informado completamente el administrador.

## **7. Recomendaciones.**

Este proyecto busca dar a conocer y ver una realidad que pasa en los conjuntos residenciales de muchos sectores en Bogotá Colombia, el cual es la falta de comunicación entre el administrador y propietarios por el factor tiempo de disponibilidad de ambas partes, el cual el administrador es una persona con un papel fundamental de administrar los recursos físicos y económicos que se dan mensualmente, por esta razón este software busca que todas las personas residentes de un conjunto se concienticen que debe haber una comunicación entre ambas partes para decidir cómo va a ser o que se va a hacer en el espacio que habita. Por medio de SIACC se evidencia un cambio de paradigma y es el de hallar la comunicación por medio de una herramienta Web la cual hace beneficiaria tanto al residente como al administrador y evita la no participación de decisiones o la desinformación de su espacio de convivencia por parte del residente y la ayuda a recibir ideas e información la cual ayuda a tomar decisiones por parte del administrador.

## 8. Bibliografía.

### ➤ Textos y publicaciones

Roger S. Pressman, Ingeniería de software, un enfoque práctico, Sexta edición, Mc Graw Hill.

Thierry GROUSSARD, Java Enterprise Edition - Desarrollo de aplicaciones web con JEE 6, eni ediciones.

### ➤ Infografía.

**AJAX**, FC: 21/05/2011,[en línea], disponible en internet, URL:  
<http://es.wikipedia.org/wiki/AJAX>

**APACHE TOMCAT**, FC: 26/11/2011, [en línea], disponible en internet, URL:  
<http://tomcat.apache.org/index.html>

**API**, FC: 06/11/2010, [en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Interfaz\\_de\\_programaci%C3%B3n\\_de\\_aplicaciones](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones)

**BASE DE DATOS**, FC: 17/09/2011, [en línea], disponible en internet, URL:  
<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

**BEANS**, FC: 06/11/2010, [en línea], disponible en internet, URL:  
<http://es.wikipedia.org/wiki/Bean>

**CICLO DE VIDA DE RUP**, FC: 19/08/2012, [en línea], disponible en internet, URL:  
<http://es.scribd.com/doc/297224/RUP>

**DAO**, FC: 06/11/2010, [en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Data\\_Access\\_Object](http://es.wikipedia.org/wiki/Data_Access_Object)

**DESARROLLO EN JEE6**, FC: 17/09/2011, [en línea], disponible en internet, URL:  
<http://www.oracle.com/technetwork/java/javaee/overview/index.html>

**FACELETS**, FC: 11/06/2011, [en línea], disponible en internet, URL:  
<http://programacionconejemplos.blogspot.com/2011/08/que-es-facelets-52.html>

**FRAMEWORKS**, FC: 21/05/2011,[en línea], disponible en internet, URL:  
<http://es.wikipedia.org/wiki/Framework>

**GUI**, FC: 09/04/2011,[en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Interfaz\\_gr%C3%A1fica\\_de\\_usuario](http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario)

**HTML**, FC: 06/11/2010, [en línea], disponible en internet, URL:  
<http://www.ri5.com.ar/ayuda03.php>

**IMAGEN SCRUM**, FC: 19/08/2012, [en línea], disponible en internet URL:  
[http://es.wikipedia.org/wiki/Archivo:Ficha\\_scrum.png](http://es.wikipedia.org/wiki/Archivo:Ficha_scrum.png)

**JAVA**, FC: 06/11/2010, [en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

**JAVA**,FC: 06/11/2010, [en línea], disponible en internet, URL:  
<http://www.todoexpertos.com/categorias/tecnologia-e-internet/programacion/java/respuestas/1737936/licencia-java>

**JEE 6**,FC: 17/09/2011, [en línea], disponible en internet, URL:  
<http://www.librosaulamagna.com/libro/JAVA-ENTERPRISE-EDITION.-Desarrollo-de-aplicaciones-web-con-JEE-6/347379/8695>

**JPA**, FC: 09/04/2011,[en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Java\\_Persistence\\_API](http://es.wikipedia.org/wiki/Java_Persistence_API)

**JSF**, FC: 06/11/2010, [en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/JavaServer\\_Faces](http://es.wikipedia.org/wiki/JavaServer_Faces)

**JSP**, FC: 09/04/2011,[en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/JavaServer\\_Pages](http://es.wikipedia.org/wiki/JavaServer_Pages)

**JUDE**, FC: 26/11/2011, [en línea], disponible en internet, URL:  
[http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CH0QFjAB&url=http%3A%2F%2Fpetra.euitio.uniovi.es%2F~i1667065%2FHd%2Fdokumentos%2FHerramientasOO.doc&ei=PggfUNK\\_DeLY0QGrzYGwDg&usg=AFQjCNH-RytiXDvf\\_\\_IlbLaDRzYKdxGoYQ&sig2=b610gPOjDOyX1929xmLH7Q](http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CH0QFjAB&url=http%3A%2F%2Fpetra.euitio.uniovi.es%2F~i1667065%2FHd%2Fdokumentos%2FHerramientasOO.doc&ei=PggfUNK_DeLY0QGrzYGwDg&usg=AFQjCNH-RytiXDvf__IlbLaDRzYKdxGoYQ&sig2=b610gPOjDOyX1929xmLH7Q)

**MVC**, FC: 13/08/2011,[en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

**NETBEANS**, FC: 21/05/2011,[en línea], disponible en internet, URL:  
<http://netbeans.org/features/index.html>

**PERSISTENCIA**, FC: 09/04/2011,[en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Persistencia\\_de\\_objetos](http://es.wikipedia.org/wiki/Persistencia_de_objetos)

**POSTGRESQL**, FC: 06/11/2010, [en línea], disponible en internet, URL: <http://es.wikipedia.org/wiki/PostgreSQL>

**POSTGRESQL**, FC: 26/11/2011, [en línea], disponible en internet, URL: <http://www.postgresql.org/about/press/presskit83.html.es>

**PROPIEDAD HORIZONTAL**, FC: 06/11/2010, [en línea], disponible en internet, URL: [http://es.wikipedia.org/wiki/Propiedad\\_horizontal](http://es.wikipedia.org/wiki/Propiedad_horizontal)

**RICHFACES**, FC: 06/11/2010, [en línea], disponible en internet, URL: <http://es.wikipedia.org/wiki/RichFaces>

**RUP**, FC: 19/08/2012, [en línea], disponible en internet, URL: [http://webcache.googleusercontent.com/search?q=cache:G00C7bQVA0J:bibliotec.a.usac.edu.gt/tesis/08/08\\_7691.pdf+metodologia+RUP&hl=es&ct=clnk&cd=6&gl=c&client=firefox-a](http://webcache.googleusercontent.com/search?q=cache:G00C7bQVA0J:bibliotec.a.usac.edu.gt/tesis/08/08_7691.pdf+metodologia+RUP&hl=es&ct=clnk&cd=6&gl=c&client=firefox-a)

**SCRUM**, FC: 19/08/2012, [en línea] disponible en internet URL: <http://www.proyectosagiles.org/que-es-scrum>

**SERVIDORES WEB**,FC: 26/11/2011, [en línea], disponible en internet, URL: <http://www.aranzadi.es/index.php/informacion-juridica/informacion-interes/glosario-de-terminos-sobre-internet-y-spam>

**SERVLET**, FC: 09/04/2011,[en línea], disponible en internet, URL: [http://es.wikipedia.org/wiki/Java\\_Servlet](http://es.wikipedia.org/wiki/Java_Servlet)

**SOFCART**, FC: 17/09/2011, [en línea], disponible en internet, URL: <http://www.puntoexe.com.co/web/es/sofcar>

**SOLUCIÓN**, FC: 28/01/2012,[en línea], disponible en internet, URL: <http://definicion.de/solucion/>

**TOMCAT**,FC: 26/11/2011, [en línea], disponible en internet, URL: <http://www.raulrodriguezcabestany.com/apache-tomcat>

**UML**,FC: 26/11/2011, [en línea], disponible en internet, URL: <http://www.docirs.cl/uml.htm>

**WEB SERVER**, FC: 21/05/2011,[en línea], disponible en internet, URL: [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web)

**WORKFLOW**,FC: 26/11/2011, [en línea], disponible en internet, URL: <http://www.pixelware.com/workflow-flujo-trabajo.htm>

**XML**, 06/11/2010, [en línea], disponible en internet, URL:  
[http://es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://es.wikipedia.org/wiki/Extensible_Markup_Language)

**XML**, 26/11/2011, [en línea], disponible en internet, URL:  
<http://geneura.ugr.es/~jmerelo/xml/>

## **9. Anexos**

Anexo 1. Encuesta a propietarios.

Anexo 2. Entrevista a administradores.

Anexo 3. Resultados de las encuestas a propietarios.

Anexo 4. Manual de usuario.

Anexo 5. Manual de instalación

Anexo 6. Diseño de las pruebas.

Anexo 7. Instalación Apache Tomcat – Ambiente de despliegue

Anexo 8. Pruebas de sistema y de integración.